AFRL-RI-RS-TR-2008-6
Final Technical Report
January 2008

MULTI-AGENT TECHNOLOGY FOR AIR SPACE DECONFLICTION

Advanced Technical Concepts, Inc.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-6 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/ /s/

NATHANIEL GEMELLI
Work Unit Manager

JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* JAN 08	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* Sep 06 – Sep 07

4. TITLE AND SUBTITLE MULTI-AGENT TECHNOLOGY FOR AIR SPACE DECONFLICTION	**5a. CONTRACT NUMBER** FA8750-06-C-0245
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 62702F
6. AUTHOR(S) Victor Skormin	**5d. PROJECT NUMBER** 558B
	5e. TASK NUMBER 06
	5f. WORK UNIT NUMBER 01
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Advanced Technical Concepts, Inc. 352 Ford Hill Road Berkshire NY 13736	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RISB 525 Brooks Rd Rome NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TR-2008-6

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-0029

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The objective of this effort is to develop computer-based multi-agent system technology that will enhance airspace control. It is a project specifically dedicated to the problem of air traffic control within airport air space in emergency situations when a hijacked aircraft appears and operates in the airport air space while ignoring the safety provided by air traffic control rules and air traffic operator commands thus providing significant threat to the “normal aircraft”.

15. SUBJECT TERMS
Multi-agent negotiation, airspace de-confliction, predictive analysis

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Nathaniel Gemelli
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UL	67	**19b. TELEPHONE NUMBER** (*Include area code)* N/A

Table of Contents

Executive Summary

This Project is a continuation of the previous one entitled "Multi-agent Technology for Airspace Deconfliction" performed according to the Extension 3 to the Partner Project of EOARD–ISTC # 1993P (2000-2005). According to our best knowledge, the latter was practically the first Project specifically dedicated to the problem of air traffic control within airport air space in emergency situations when a hijacked aircraft appears and operates in the airport air space while ignoring the safety provided by air traffic control rules and air traffic operator commands thus providing significant threat to the "normal aircraft".

A novelty of the problem statement as well as its difficulty to solve is caused by highly dynamical changing of the air traffic control situations when the air traffic operator is not able to cope, in real time, with providing safety and security of the "normal" aircraft simultaneously operating within airport airspace. Due to novelty of the problem, the former Project objectives were the initial study of the problem in question peculiarities, understanding its key issues and subtasks as well as investigation of the possibilities and limitations of the automated autonomic air traffic control fulfilled by negotiating software agents assisting the pilots of "normal" aircraft with minimal intervention of the air traffic operator. Accordingly, at that stage and due to very short term of the research (5 months) the problem was stated in a simplified form.

In the Project reported, main part of the simplifications assumed in the earlier one is omitted while providing much more real life problem statement as well as real life research objectives.

The formal contract was signed on September 21, 2006 and the research was started from October 5, 2006. The Project Work Plan is divided in two phases. The first phase is scheduled for 12 months, from October 1, 2006 till September 30, 2007. Particular research efforts planned for the *first phase* according to the Contract are formulated as follows:

Task 1. Acquisition of numerical and factual information representing particular components of the airspace deconfliction task environment: weather conditions, airport and adjoining airspace topology, and scheduled air traffic. Development of data structures for storing the above mentioned information that would support the most efficient user interface with the purpose of editing and retrieval of the information.

Task 2. Development of a realistic conceptual model of the safe air that has to provide the aircraft motion within given boundaries and meeting separation requirements given in terms of minimum allowable distance between aircraft.

Task 3. Development of typical deconfliction situations addressing a variety of behavior patterns of the hijacked aircraft, the air traffic configurations, and weather (visibility) conditions.

Task 4. Development of an algorithm of airspace deconfliction, addressing the specific conditions of the involved aircraft represented by autonomous software agents by incorporating a trade-off negotiation within the agent community.

Task 5. Software implementation of the deconfliction procedure, verification, and validation of the procedure as applied to different deconfliction situations.

Task 6. Development of a multi–agent airspace deconfliction system architecture and protocols supporting interaction of distributed entities (agents) routinely participating in the application of the deconfliction procedure.

Task 7. Development of a graphical user interface communicating computer generated airspace deconfliction decisions to the participating pilots and air traffic controllers.

Task 8. Development of the particular components of the deconfliction system software prototype.

Research experience during the reported phase showed that the deconfliction task cannot be modeled and solved with ignorance of model and algorithms of air traffic control in "normal" situation when aircraft carefully follow the schedule, air traffic rules and commands of air traffic operator. This task is primarily intended for providing safety of aircraft, and when hijack appears all "normal" aircraft set up initial air traffic configuration that, further on, has to be deconflicted using new separation standards (increased around the hijacked aircraft) and weakly predicted model of behavior of the latter. This means that air

traffic control in a state of emergencies (hijacking, as well as aircraft's technical faults, etc.) should be considered as a particular (of course, very important, difficult and specific) case of air traffic control when different safety policies are used. Therefore, the basic principles of air traffic control in "normal" and emergency situations should remain the about same. The air traffic control system should be provided by adaptive behavior concerning re-allocation (when necessary) of responsibilities between air traffic control operator and airborne software means while implementing autonomous behavior of "normal" aircraft. The main roles, in providing such behavior, agent-based intelligent software assisting the pilots and operator(s) have to belong.

Thus, to achieve the Project objective, it is necessary to design and implement system components responsible for air traffic control in "normal" situations. Of course, this task leads to the necessity to use a more general statement of the air space deconfliction problem, while extending it with the aforementioned task. This is the reason why so much attention is below paid also to the air traffic control task in "normal" situations.

Chapter 1 provides brief introduction into the results of the former Project, describes differences and interconnections between the problems statements peculiar to previous Project and reported one with the focus on omitted assumptions making the problem statement much closer to the reality.

Chapter 2 summarizes numerical and factual information representing particular components of the airspace deconfliction task environment and corresponding data structures assumed by the Task 1 of the Work plan.

Chapter 3 presents the developed realistic conceptual model of the safe air that has to provide the aircraft motion within given boundaries and meeting the separation requirements given in terms of minimum allowable distance between aircraft. This result corresponds to the solution of the Task 2 of the Work plan. It also describes typical behavior patterns of normal and hijacked aircraft and air traffic configurations to be modeled in the Project as well as organizational structures of air traffic control that are assumed by the tasks 3.

Chapter 4 describes developed airspace deconfliction algorithm addressing the specific conditions of the involved aircraft whose pilots are assisted by autonomous software agents. These agents provide distributed autonomous decision making implementing cooperation through trade-off negotiation within their community. Solution of this task is assumed by Task 4 of the Project Work plan.

Chapter 5 carefully describes the developed design project of multi-agent airspace deconfliction system, specification of its basic components and their interaction. It includes specification of the multi-agent airspace deconfliction system meta–model and protocols representing architecture of the system in question (assumed by Task 6), model of the simulation server that has been used for verification and validation of the software implementation of the developed deconfliction algorithm (Task 5) and particular multi-agent airspace deconfliction system components (Task 8).

Chapter 6 presents graphical user interface providing visualization of the air traffic configurations and corresponding situations in both, normal situations when only "normal" aircraft operate within airport airspace and abnormal ones, when a hijacked aircraft is operating as well. This interface corresponds to the solution of the Task 7 of the Project Work plan. At the same time, this interface plays the role of important component of the software means supporting verification and validation of the airspace deconfliction algorithm itself.

It can be noted that the order in which the results of the research are presented in the Report is other than one assumed by the Project Work plan. The reasons of this are twofold. On the one hand, to cope with the project work plan it was necessary to solve an additional task, development of multi-agent air traffic control in "normal" situations that was not assumed by Work plan. On the other hand, a discrepancy between task ordering assumed by Work plan and ordering of the corresponding materials in the Report is caused by the natural logic of the research (sequencing of the tasks) appropriate for the authors. In practice, practically all the tasks are strongly correlated and sometimes indivisible. The order of the task-related results used in the Report found out the most appropriate for the authors.

In general, all the tasks assumed by the Project Work plan are solved.

1. Introduction

1.1. Project Starting Point: Experience Accumulated in Previous Project

The short term project "Multi-agent Technology for Airspace Deconfliction" (July 1–November 30, 2006, EOARD-ISTC #1993P, Extension 3) was, at least for research group of SPIIRAS, the first Project specifically dedicated to the problem of air traffic control within airport air space in emergency situations when a hijacked aircraft appears and operates in the airport air space while ignoring the safety provided by air traffic control rules and air traffic operator commands thus providing significant threat to the "normal aircraft". The results of this project are presented in [1993-Task 1-Addendum 3]. The former project, in turn, was essentially based on the theoretical, architectural and technological results received in the process of the research on the main project #1993PP performed since 2000 ([1993P Task 1-2003], [1993-Task 1-Ext 1-2 2005]).

The results obtained in the project "Multi-agent Technology for Airspace Deconfliction" are as follows:

- A typical structure of an airport airspace that was then used as a case study for justification of the main conceptual and design solutions concerning multi-agent airspace deconfliction system was developed. This structure was specified formally in terms of Scenario Knowledge Base framework providing a number of useful properties of the resulting formalization. The most important properties are automatic satisfaction of constraints imposed by airport airspace structure which are resulted from *adaptation and simplification* of the rules and the regulations providing safety of the air traffic control. This structure and its formal specification were further used as a testbed for study, investigation, verification and validation of various airspace deconfliction algorithms limited by the admissible movements of the "normal" aircraft set by this structure.
- Reasonable assumptions and simplifications of the airspace deconfliction problem and problem statement were developed. The proposed problem statement made it easier the general study and investigation of the problem peculiarities and deconfliction algorithms properties.
- Typical scenario of airspace deconfliction task within homeland security scenario determining basic entities involved in distributed task solving, their roles and necessary coordination of their distributed performance forms constituting conceptual basis for the task decomposition and its design and implementation within multi-agent framework was developed.
- Formal model of the airport airspace structure specified in terms of Scenario Knowledge Base and formal model of constrained movement of the aircraft subjected to the constraints determined by the aforementioned airport airspace structure were developed. These models were used as a basis for development and efficient implementation of the deconfliction algorithm represented in terms of notions of these simplified models.
- Distributed algorithm of airspace deconfliction specifically designed for multi-agent implementation was developed, implemented and verified. This algorithm may be considered as a first version making it possible to study deconfliction task high level properties that provided a basis for building real-life airspace deconfliction algorithms.
- Formal specification of the meta–model of multi-agent airspace deconfliction system representing formally the roles of the system distributed entities, protocol of their interactions and messages, with which they exchange was developed. As well, formal specification of services provided by agent classes proposed in the designed project of multi-agent airspace deconfliction system which were represented in terms of state machines were developed. Both these formal models constitutes design project of multi-agent airspace deconfliction system which was implemented and tested in simplifies version.

1.2. Comparison of the Project-2006 and the Reported One

Let us assess the simplifications assumed by the problem statement in the reviewed project contrasted to the assumptions used in the reported one. In short, this comparison is presented in Tab. 1.

Analysis of the Tab. 1 content shows that the most of simplifications and assumptions are either omitted or significantly weakened. New conceptual model of aircraft' movement meets real life practice. It assumes to model movements of aircraft according to their full technical capabilities, including *Off-path jogging* maneuver, coming up with other aircraft in order to meet separation standards and temporal constraints when necessary according to schedule, etc. As a result, each leg may be gone through for varied time what provides the air traffic control system with new dimension of controllability (and complicates the system development accordingly).

Table 1.1. Contrasting assumptions and simplifications supposed in twp sequential projects

#	Project as of 2005	Project as of 2006-2007
	Simplifications and Assumptions	
1.	All aircraft including hijacked one have the same constant velocity and rate of climb capabilities	*Omitted.* The aircraft may fly with various velocities according to their technical capabilities. Additionally, an aircraft velocity depends on the altitude it is flying
2.	Each leg, holding zone and landing circle in the airport area airspace is assigned a time interval an aircraft spends while moving from its entry point to exit one.	*Omitted* due to omitting of the previous simplification
3.	The hijacked aircraft can move along any path and at any height and does not agree its movement with the air traffic dispatcher, but in airspace deconfliction it is assumed that in the future the hijacked aircraft will be moving rectilinearly and uniformly.	*Held*. But the variety of the hijacked aircraft behavior patterns is expanded significantly
4.	Each leg and orbit are determined by coordinates of entry and exit points of its central line.	*Weakened.* An aircraft may hold any position within leg or holding circle.
5.	The prohibited area around the hijacked aircraft which normal aircraft have to leave as soon as possible is determined as follows: the space within the rectangular tube of given horizontal and vertical sizes (e.g. 10 miles and 500 m. respectively) and situated ahead of hijacked aircraft up to fixed length (for example, the length of 10 miles).	*Weakened*. The separation standards are determined much more flexible and are determined by the safety and security policies depending on various factors and specified in terms of rules which truth values are instantiated depending on situation attributes.
6.	Transition of a normal aircraft from its current position (state, node) to other ones is forbidden if the target position: • is occupied currently by other aircraft; • is overlapping with the prohibited area; • is forbidden by the airport area topology	*Omitted.* Permitted and prohibited transitions of normal aircraft from current positions (state) are determined by safety and security policies. The notions of airport air space "structure" and "node" are not used (in the ontology).
7.	Planning, scheduling, plan performance, etc. are *external event-driven*.	*Omitted.* Only event corresponding to the fact of appearance of the hijacked aircraft and its disappearance are valid, while switching the safety and security policies to the relevant section of rules

8.	Time interval needed for agents providing pilots with airspace deconfliction assistance is negligibly small in comparison with the time between events implying re-planning of airspace deconfliction. The validity of this simplification was not checked.	*Held*. The admissibility of this simplification is a subject for verification via simulation when a software prototype of multi-agent airspace deconfliction system is developed (planned to the end of the second phase research).
9.	Hijacked aircraft is not aware of positions and courses of other aircraft operating within airport airspace.	*Omitted*, but there is no scenarios in which this knowledge is used by hijackers.
10.	Movement of aircraft in the outer space is not considered	*Omitted*. For aircraft arriving to the airport air space the time of entrance and entry sector is predicted
11.	The aircraft arrive from outer airspace to the airspace of the airport area through given entry points (from fixed sectors of directions).	*Held*
	Only one aircraft may be situated in each particular leg, holding zone, landing circle and on runway.	Omitted. These situations are managed by safety policy.
	Airport has two runways	Omitted. Airport may have arbitrary count of runways.
	Fuel content of each aircraft including hijacked one is given in terms of time it can fly safely.	Omitted. .
	Hijacked aircraft (HA) can follow along any trajectory.	Hold

Although in the previous project there were no special constraints on the behavior patterns of the hijacked aircraft, only the case of uniform movement with the constant course and velocity was practically simulated. If to take into account the diversity of the hijacked aircraft behavior pattern the task becomes closer to real life situations and, at the same time, more complicated to model and implement. Exactly the last case is the subject of modeling and future implementation in the reported research.

The Project as of 2005 considered movement of the aircraft precisely along with the central line of a leg or circle. This research weakens this requirement while permitting to aircraft to follow along any trajectory within space assigned to legs or holding circles.

An important peculiarity of the present research is that the separation standards are considered as dependent on situations and strictly formulated in terms of safety and security policies. In contrast, in the previous research, separation standards were formulated in terms of the airport airspace structure elements independently of their length. As a result, while using such a safety policy, the airport airspace was utilized much less effectively, but the task of providing meeting of the separation standards was a simpler task as compared with this task of the reported project. Fortunately, in the last case the airport airspace may be utilized much more effectively.

The model designed in the reported research is driven mostly by internal events produced by interacting agents assisting the pilots as well as air traffic operator(s). In the previous research, such model was primarily external event driven. The former variant of modeling requires more intelligent air traffic control but it corresponds to more autonomy of pilot assisting agents than in the last case.

An important step ahead is made in current project due to the fact that, in it, the aircraft approaching to the airport but not reached yet to the arrival zone are also the subjects of planning, scheduling and deconfliction through prediction the time of their appearance in arrival zone. The expansion of the area of

air traffic control males it possible to model continuous control process that exactly corresponds to real life situation.

Additionally the previously accepted limit of the runway count is also omitted although in the case study used in the reported research the JFK airport is considered where exactly two runways exists.

Practically, many other novelties are modeled in this research. Among them, real life airports may be used as case studies, real life data structures are used for representation of some aspects of the problem domain, etc. All this peculiarities will be highlighted below.

As a conclusion, it can be said that the problem statement as well as approaches and models developed within this project crucially distinguish in comparison with the ones of Project as of 2005 and in the main respects correspond to the real life cases.

2. Numerical and Factual Information Representing Particular Components of the Airspace Deconfliction Task Environment and Corresponding Data Structures

This Chapter presents the materials assumed by the task 1 of the Work plan on the contract. It presents conceptual description of basic notions used for specification of the actual environment state as well as data structures used for storing the aforementioned information in data bases of airspace deconfliction system that is the target of the research.

The above mentioned notions, types and representations of them in data bases as well as their concrete values were resulted from the following sources:

1. Official documents issued ([ARINC-424], [ICAO Doc.4444]);
2. Domain experts on air traffic control that are the specialists from St. Petersburg University of Civilian Aviation, Department of Air Traffic Control and textbooks issued by this department;
3. Files and data structures of Microsoft Air Flight Simulator game ([MS Flight SDK]);
4. Recent scientific publications ([Tomlin et all, 1998], [Hill et all, 2005], [Tumer et all], [Tozicka et all, 2007], etc.)
5. Materials publicly available in the Internet ([Airport JFK])

The materials given below resulted from the study of the aforementioned sources and its generalization and summarization.

2.1. Airport Airspace Topology

The high level notion "*Airspace topology*" is intended to specify, in a standard way, i.e. independently of particular airports, admissible movements (trajectories) of aircraft jointly operating within airport airspace in terms of lower level notions. It turn, the latter notions are also used as arguments (attributes, data structures) for specification of the safety and security policies used by the aircraft as air traffic control parameters, as input data for deconfliction algorithms, etc. It is worth to note that *airspace topology* does not concern the air traffic configuration, i.e. current positions, speeds and courses of particular aircraft operating within airport airspace at current time instant. *Airspace topology* is a high level abstraction imposing the *basic constraints on geometry* of the admissible trajectories of aircraft presented in airspace. Let us note that, in contrast, the safety and security policies impose additional constraints on dynamics of mutual movements of aircraft meeting the geometrical constraints.

2.1.1. Basic Low Level Notions and their Representation Structures

Point of the Air Space[1]

Attribute	Comment
FixIdent	Point Identifier
Name	Point name (usually used for semantically simpler interpretation)
Lat, Long	Point's coordinates
fixType	Point Types • Waypoint – determining an orientation in the airspace • Runway – self–explaining • {VOR, NDB} – radio beacons
If fixType=runway (optional)	
airportIdent	
Number	
Designator	

[1] The table lines given in grey color corresponds to the attributes that, in current version, are out of use

Runway (Take-off and landing strip)

Attribute	Comment
airportIdent	Airport identifier
Number	Runway number
Designator	Runway orientation
fixIdent	Runway input point identifier
Alt	Absolute altitude (above see level)
Length	Runway length
Direction	Runway orientation of in degrees
Takeoff	
Landing	
aircraftType	Types of aircraft admitted for runway use

Leg – leg

Attribute	Comment
fixIdent	Leg exit point
Altitude	Altitude value of the point passing
altitudeDescriptor	• + "*At or above*" altitude specified in the field *Altitude* • - "*At or below*" the altitude specified in the field *Altitude* • (blank) "*At*" the altitude specified in the field *Altitude*. • "*At or above or below*" the altitudes specified in the field *Altitude*" and *Altitude2*
Altitude2	value
Type	Leg type • *IF* - input point • CF – course in movement to the input point • *TF* – vectoring is admissible (from point to point) • *HF* – waiting orbit starting from the input point
If type = HF	
Distance || Time	Length of holding orbit or time of its passing
turnDirection	New direction after turning
If type = CF	
Direction	

2.1.2. Movement Schemes of Aircraft within Airport Airspace

Arrival scheme

Attribute	Comment
fixIdentArrival	Arrival identifier
fixIdentInitial	Arrival point identifier
fixIdentTo	Identifier of the last arrival point (in arrival zone)
ArrivalsLegs	Arrival scheme - sequence of legs usage

Approach scheme

Attribute	Comment
fixIdentRunway	Runway identifier
fixIdentInitial	Identifier of the entry point
ApproachLegs	Approach scheme - sequence of legs usage
typeNAV	Navigation system type
missedAltitude	Altitude of missed approach movement
MissedApproach	Missed approach scheme - - sequence of legs usage

Transition scheme

Attribute	Comment
fixIdentFrom	Initial point
fixIdentTo	Destination point
TransitionLegs	Sequence of legs usage

Fig. 2.1 and 2.2 exemplify airspace topology (in horizontal and vertical projections respectively) within New York city including three airports, – JFK, La Guardia, Republic. Fig. 2.3 depicts the approach zone of JFK airport.

In general words, the airport airspace topology is divided into two zones: (i) arrival zone and (ii) approach zone.

Arrival zone is divided into *Arrival schemes*, for instance, in Fig. 2.1, nine arrival schemes are presented. Each arrival scheme begins at entry points to airport airspace. It is specified as a sequence of legs ending with *holding area*.

Approach zone comprises *approach schemes*. These schemes are not depicted in Fig. 2.1 due to too small scale. Each approach scheme begins at entry point (into approach zone), consists of sequence of legs and completes with a runway of the airport.

Movement schemes within approach zone can be classified in two categories (with some uncertainty) that are (i) standard approach schemes and (ii) *missed approaches* schemes where the latter correspond to the cases if some non-standard situation occurs (technical problems, air traffic control unpredictable situation, hijacking, etc.), As a rule, missed approach cases result in the necessity to use holding orbits. Transition schemes bind the destination points of the arrival schemes and entry points of approach schemes. As a rule, each arrival scheme is bound with several approach schemes. In general case, transition schemes are also used for binding of different arrival schemes.

Fig. 2.2 depicts movement schemes (arrival and departure) projected onto vertical plane. In the left part of the figure, along vertical axis, the echelon scale (from 0 till 30,000 feet with quantization step equal to 1000 feet) is presented. In this figure, the vertical projection of an arrival scheme and approach scheme passing through SHANK, FRILL, etc. points is given in red color.

Specification of the airport airspace topology determines also admissible echelons, i.e. admissible altitudes of passing of the legs exit points of type IF, CF, TF. For instance, while passing through the SHANK point, aircraft are admitted to use the echelons in between 24, 000 – 30,000 feet. If the leg is of type either CF or TF then its exit point may be bound with holding area. In particular, all leg legs of arrival scheme, depicted in Fig2.2, excluding leg identified as CCC, are bound with holding area

Airport airspace topology specification contains also the departure schemes. They begin at a runway and end at a point of airport airspace leaving. Since, according to aircraft' performance characteristics, the climbing rate, as a rule, exceeds its descending rate, the aircraft' exit points are located, in projection of departure trajectory onto horizontal plane, in between approach zone and arrival zone boundaries.

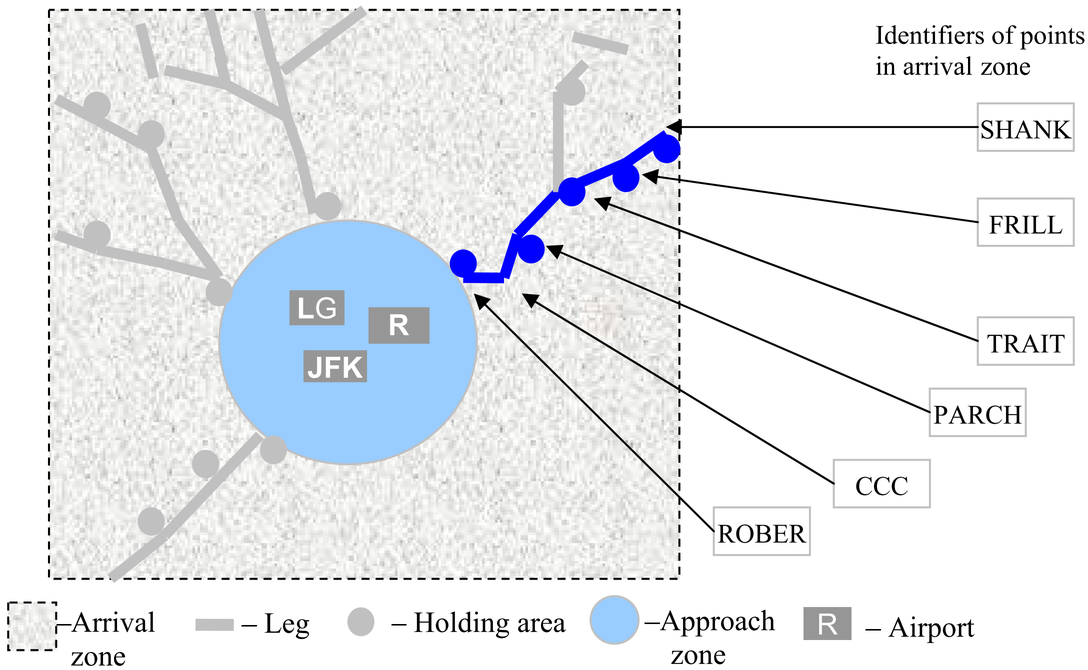


Fig.2.1. Airspace topology within the New York City area

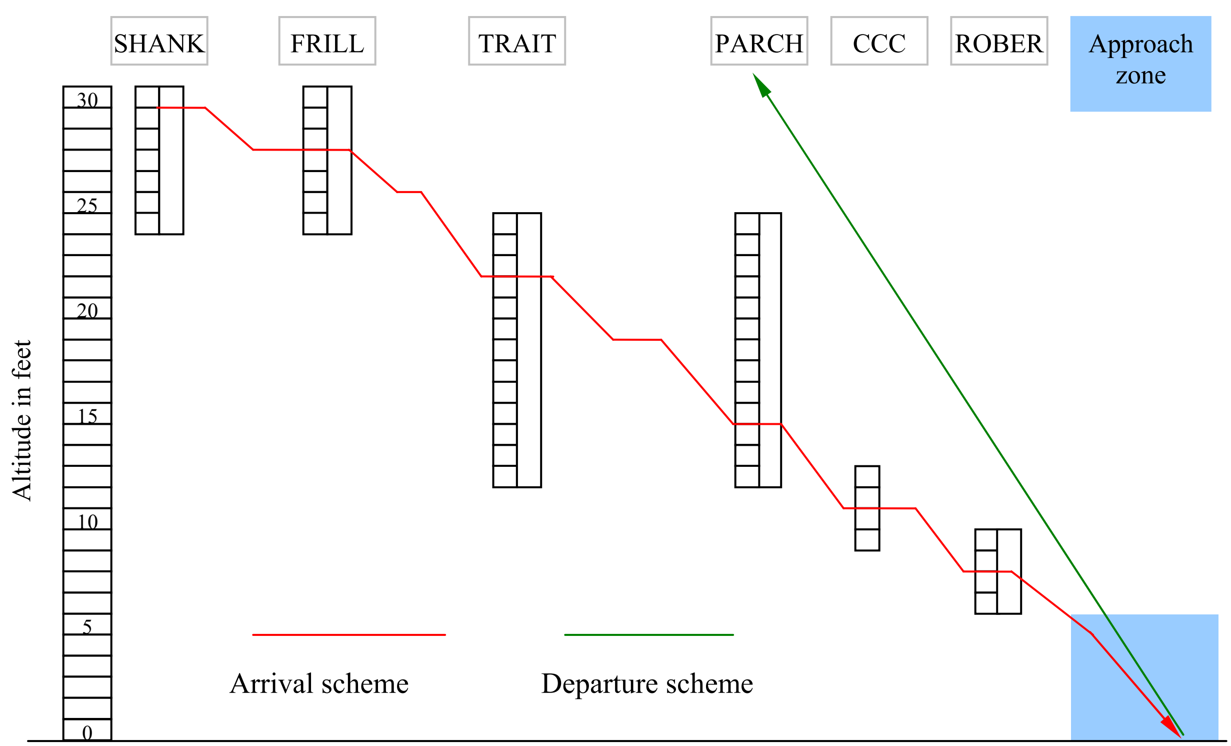


Fig.2.2. Representation of arrival and departure schemes in vertical projection

2.2. Aircraft Classification

Classification of aircraft is an important issue due to their different characteristics, requirements they put to the landing ad take off facilities, etc. Indeed, different aircraft may have different taking off and landing speed and length, different climbing and descending rates, different navigation equipment, different weights, what is very important in air traffic planning and control. In this research, the classification given in Tab. 2.1 is used.

Fig.2.3. Approach zone of JFK airport

Table 2.1. Aircraft classification

Aircraft classes	Speed limits within arrival zone depending on altitude км/per hour (min-norm-max)		Speed limits within approach zone (circle zone), depending on altitude км/per hour (min-norm-max)		Cruising speed (at 27000 feet)	Horizontal acceleration, км/per hour per second	Landing speed
	18000 feet	9000 feet	3000 feet	1800 feet			
1	700-800-900	500-610-750	310-420-530	270-350-500	850-950	6-10	240-345
2	580-680-750	450-540-600	300-400-520	260-330-400	650-850	4,5-8	215-295
3	420-450-480	400-450-480	300-380-450	220-270-320	450-550	2-3	150-140
4	280	280	200-240-270	180-210-250	180-280	2-,2,5	130-185

Note: 1. In the intermediate altitudes, the speed limits are calculated via linear interpolation.

Other characteristics of the aircraft are also important. For example, the attributes of the aircraft like length of take off and landing distances, weight, maximal acceleration, rates of landing and take–off speeds climbing
and descending, and some others are very interdependent and also influence in different ways on air traffic planning. But in this research the above mentioned dependencies are ignored in air traffic planning and scheduling task thus forming a number of simplifications. Of course, they may influence on the resulting plan and schedule of air liners and on quality of deconfliction. Nevertheless, they are used below and the following arguments justify admissibility of them within current research:

1. The main objective of the research is multi-agent algorithm and technology for airspace deconfliction and it is most probable that these simplifications are not crucial in regard to the algorithm itself.
2. These simplifications make modeling of the aircraft movements much easier thus helping to save the total efforts to be spent for secondary task while concentrating on distributed deconfliction algorithm.
3. These simplifications may be omitted when necessary or if the customer believes they are too hard.

4. Development of a realistic conceptual model of the safe air that has the provision for variable speed of the aircraft motion within given boundaries and separation requirements given in terms of minimum allowable distance between aircraft.

2.3. Air Traffic-related Situation Model

2.3.1. Schedule of aircraft arrival–departure

In airspace deconfliction task, it is assumed that before the time moment when a hijacked aircraft appears in the airport airspace, the air traffic control is being performed in normal mode, i.e. arrivals and departures of aircraft is being handled according the normal schedule assumed for the airport. That is why, for normal air traffic situation, its arrivals and departures schedule determines the air traffic–related situation. This information added with the data representing the classes of aircraft and their characteristics (see section 2.2) forms the input data of the air traffic-related situation model.

Information specifying each arriving aircraft is composed of the following components:

- Aircraft class;
- Entry point of aircraft into airport airspace;
- Altitude of aircraft entry point;
- Time of entry;
- Destination runway.

Analogous information concerning departing aircraft is of the following format:

- Aircraft class;
- Take-off runway;
- Take-off time according to the schedule;
- Exit point of aircraft from airport airspace;

2.3.2. Weather Conditions

Weather conditions impose some additional limitations on use of some elements of the airport airspace topology and airport runways. If a side wind exceeds the given threshold then utilization of some runways is prohibited for some or all classes of aircraft. As a result, some approaching elements of the airport airspace topology are closed. Presence of thunderous clouds within airport airspace may lead to the situations when some arrival and/or departure schemes can be forbidden to use.

2.4. Chapter concluding comments

This Chapter presented the results of the research on Task 1 of the Project Work plan.

3. Realistic Conceptual Model of Safe Air Traffic

3.1. Separation Standards

3.1.1. Mutual Behavior Patterns of Pairs of Normal Aircraft and Attributes Determining Safety

Separation standards defined for various air traffic–related situations (relations between "normal" aircraft) form the basis of safety of air traffic while determining corresponding situation-based safety policy. Let us consider the separation standards and then formulate rule-based safety policy to which the aircraft have to follow within the airport airspace.

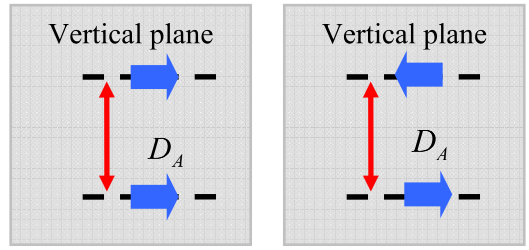


Fig. 3.4. Distance D_A

Horizontal movements of aircraft occupying different echelons

An attribute that determines minimal admissible *vertical distance* between pair of "normal" aircraft if they are flying strictly horizontally is further denoted by the symbol D_A (Fig. 3.4).

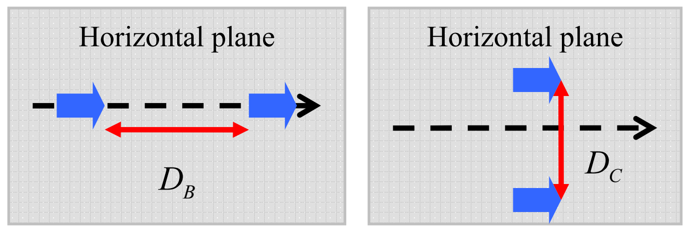


Fig. 3.5. Distances D_B and D_C

"Following" motions of aircraft within the same echelon of altitude

The attributes determining separation standards for this case are

D_B – minimal longitudinal distance measured along the axis line of the legs (Fig. 3.5) and

D_C - minimal distance between the trajectories of aircraft measured in orthogonal directions to the longitudinal axes of aircraft.

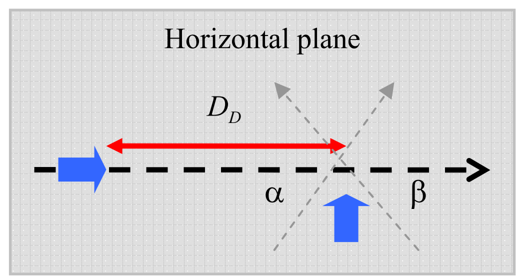


Fig. 3.6. Distance D_D

Transverse motions of aircraft occupying the same altitude echelon

It is said that the aircraft are moving along the cross–cut trajectories if the angle value between the trajectories in horizontal plane is more than 70 and less than 110 degrees (see Fig. 3.6). The attribute determining the separation distance between the aircraft is denoted as D_D. It is the distance from aircraft to the trajectories crossing point when one of the aircraft has achieved the crossing point.

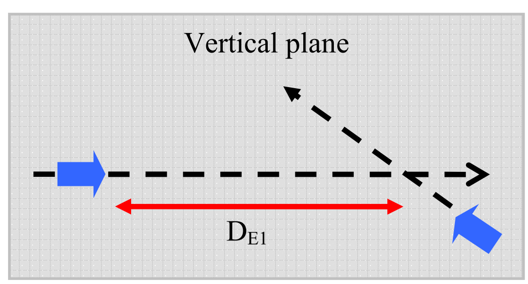


Fig.3.7. Distance D_{E1}

Head motion of aircraft one of which is changing altitude echelon

It is said that aircraft have "*head motion*s" if one of the aircraft is moving horizontally while the second one is climbing or descending with a vertical speed V_A, at that the angle between the course of horizontally flying aircraft and projection of the course of other aircraft onto horizontal plane is more than 110 degrees. The distance D_E corresponds to the horizontal distance between the aircraft when one of them has achieved the trajectories crossing point. Two cases has to be distinguishes here: (1) the aircraft earlier achieving the crossing point is one changing echelon; (2) the aircraft earlier achieving the crossing point is one flying horizontally. The difference between these cases is that D_E in the fist case has to be greater than in the

second case. Let us denote the corresponding values of D_E as D_{E1} and D_{E2} respectively (Fig. 3.7 and 3.8 respectively). It is important to note that admissible values D_{E1} and D_{E2} depend on the vertical speed V_A of the aircraft changing the echelon.

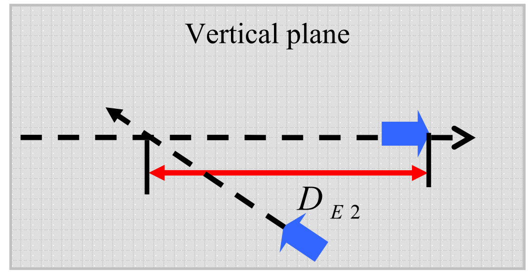


Fig. 3.8. Distance D_{E2}

The admissible values of distances D_A, D_B, D_C, D_D, .D_{E1} and D_{E2}, in general case, depends on different air traffic–related situation attributes. In the multi-agent airspace deconfliction system software prototype that is under development the following admissible values of the aforementioned distances are used:

- D_A = 0, 300 km;
- D_B = 10 km in the arrival zone and 5 km in approach zone;
- D_C = 10 km in the arrival zone and 5 km in approach zone;
- D_D = 20 km in the arrival zone and 10 km in approach zone;
- D_{E1} = 30 km if $V_A < 10$ m/per and 60 km if $V_A \geq 10$ m/per sec;
- D_{E2} =15 km if $V_A < 10$ m/per and 30 km if $V_A \geq 10$ m/per sec.

3.1.2. Attributes Determining Safety in Presence of Hijacked Aircraft

The same attributes are used for determination (specification) separation standards between normal aircraft and hijacked one. Moreover, the same policy providing security of normal aircraft in presence of hijacked one is used. The latter is called below as "*security policy*" in order to distinguish between what below relates to the safety of normal aircraft during their mutual movements and what relates to the safety of normal aircraft with regard to the hijacked one. The main difference between safety and security policies is that the admissible values of the aforementioned attributes, in case of security policy, in this research, are increased in two times.

3.2. "Normal" Air Traffic Configuration Analysis

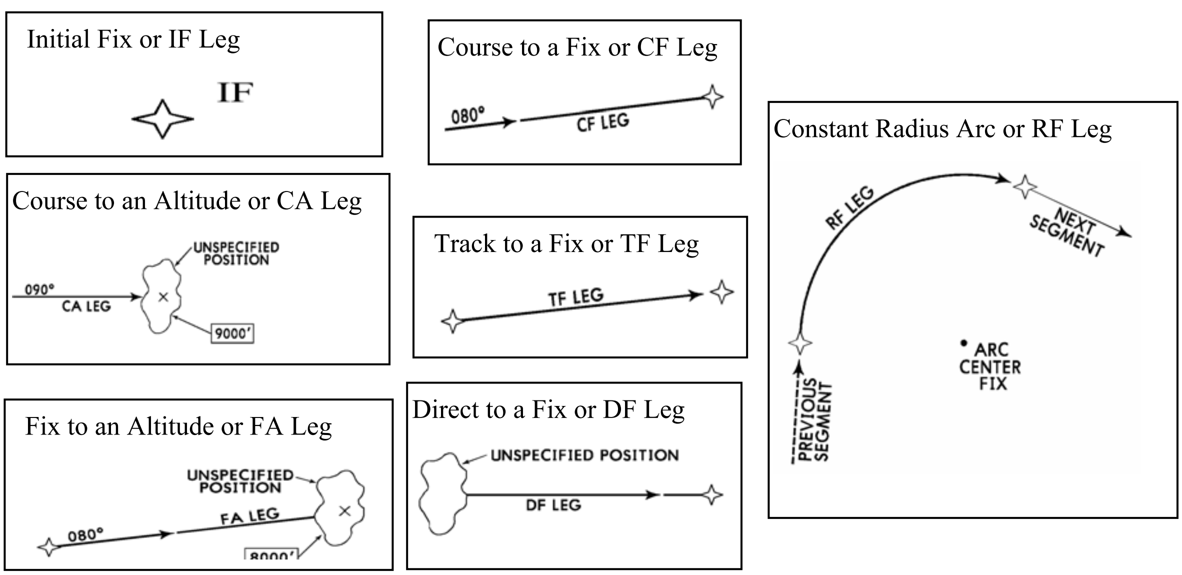


Fig. 3.9. ARINC 424 Path Terminator Concept. Selected legs examples and their graphical legend (out of 23 legs introduced). In the Project, only *IF* Leg, *CF* Leg, and *TF* Leg are used for specification of the airport topology.

3.2.1. Normal air traffic organization

Air traffic organizational principles admit some degrees of freedom for admissible aircraft movements within the arrival and approach zones while permitting, to the aircraft, some kinds of maneuvers, but "degrees of freedom" are different in these zones. Let describe the aforementioned organizational principles assumed for arrival and approach zones separately. It is worth to note that the organizational principles described below constitute the basis for development of the multi-agent airspace deconfliction system at the analysis stage.

Before analysis of normal air traffic organizational principles let us introduce what is called standard "legs" determined in the document *Supplement 19 to ARINC Specification 424: Navigation System Data Base*, their types and legends. A "leg" denotes a leg of given type. The aforementioned document determines 23 types of legs and each of them is assigned the type and legend for graphical representation. Fig. 3.9 presents some examples of standard legs and their graphical denotations. In the current step of the research, only IF Leg, CF Leg, TF Leg and HF Leg are used for specification of the airport airspace topology.

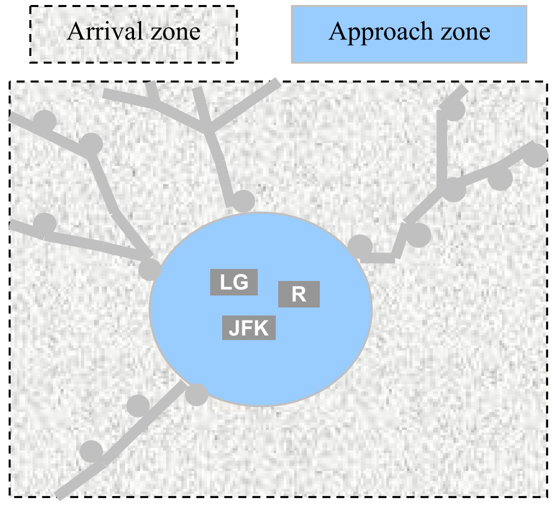


Fig.3.10. New York City airspace topology

Arrival zone (Fig 3.10)

Legs (specified by Legs of Type = CF ! TF)

- Every leg is echeloned, i.e. it consists of several sub–legs occupying various intervals of altitude;
- Trajectory of am aircraft moving along a leg may be varied around its axis line if this is necessary to hold separation standards assumed by safety policy;
- In some situations assumed by safety policy an aircraft may not follow a standard motion scheme. Such movements are called *vectoring*. Vectoring corresponds to exit out of leg with either subsequent return into it or change of the movement scheme.

Holding areas (Leg Type = HF)

- Holding areas are located within arrival zones. Occupation of a holding area is used for control of temporal attributes of arriving aircraft as well as air traffic control as whole. As a rule, holding areas are used in the cases when all the echelons of the immediately subsequent legs leading to the aircraft target are occupied by other aircraft.

Approach zone (Fig 3.10)

Legs

- As a rule, only input legs of approach schemes are echeloned. Legs which are immediately linked to the runways are determined uniquely by assigning the altitudes above every approach scheme point.
- Movement inside approach legs has to be held along its axis. In exclusive situations, e.g. when the aircraft has to "smoothly" turn, the latter is admitted to move deviate its trajectory from the leg axis. In the case mentions as example, aircraft has to move along a circular arc while simultaneously controlling the temporal attribute of the trajectory.
- In some situations assumed by safety policy an aircraft may *vectoring*. However this evolution should be carried out more carefully due to potentially high intensity of the air traffic inside approach zone.

Holding areas (holding circles)

- Inside approach zone, holding areas are presented only in the approach schemes supposing cases called "*missed approach*".

3.2.2. Some organizational principles preserving conflicting and potential conflict cases

As a rule, topology of airport airspace designed for providing safe arrival and departure of aircraft is built in a way that, for some cases, provide safety "on default", i.e. automatically. In particular, such conclusion can be inferred via joint analysis of airport airspace topology and the separation standards (see section 3.1). Practically this conclusion is predefined by corresponding organizational principles of air traffic control.

Below the aforementioned analysis is done as applied to airspace of the New York City airports used below as a case study. Of course, particular airports possess their own peculiarities from the viewpoint in question. Nevertheless, this case study is typical from the most of viewpoints.

Let us analyze, on the basis of the aforementioned case study, the organizational principles providing partially "on default" safety of aircraft movement. This issue is important because it may make it easier air traffic control and airspace deconfliction algorithms.

Arrival zone

For pair of aircraft, which a) have goal landing and b) are moving according to schemes, "*on default*" *safety* is provided in the following cases:

Case 1

The aircraft use non–overlapping schemes in arrival zones, i.e. the schemes do not have the common legs.

Case 2

The aircraft use either common or overlapping schemes but they are in not adjacent legs, i.e. in the legs that do not have common point.

Case 3

The aircraft use either common or overlapping schemes, moving through the same or adjacent legs but they are within different echelons.

A conflict *may* potentially *occur* in the following case:

Case 4

Aircraft a) have goal landing and b) move according to schemes, at that they use either common or overlapping schemes, move through the same or adjacent legs and use the same echelon. The last condition, using the same echelon, assumes that both aircraft are moving either with the zero vertical speed or one of them is moving with non-zero vertical speed while crossing the echelon of the other aircraft when the latter is moving horizontally.

Case 5

Aircraft a) have goal landing and b) one of them or both are moving out of schemes. This case corresponds to the vectoring of one or both aircraft.

Approach zone

In this zone, air traffic density is much more intensive. That is why to detect a priory conflict / conflict–free situations more sophisticated approach is needed. The main specific of this zone is that it does not contain the holding areas (holding circles) with the only exclusion concerning "*missed approach*" situation. The latter occurs only in the cases, when, for some reason, aircraft reached the approach zone but cannot land. As a consequence, the control of time is impossible inside the approach zone. The latter means that if an aircraft has entered into approach zone, the time of its flight till landing is determined only by the selected approach scheme and cannot be changed using some buffer zones.

Let us consider the formal model of the aircraft movement and air traffic task decomposition intended for detection and prevention of potential conflicts and for design a deconfliction strategy.

Let $\boldsymbol{S} = \{S_1, ..., S_N\}$ be the set of movement schemes designated in approach zone and $t(S_X)$ denotes the entry time of an aircraft into approach scheme $S_X \in \boldsymbol{S}$. Let S_A and S_B are an arbitrary pair of approach schemes belonging to the set $\boldsymbol{S}$. This pair of the schemes is called *independent* if, for any values of $t(S_A)$ and $t(S_B)$ of entry of any two aircraft using the approach schemes S_A and S_B violence of separation standards for these aircraft is impossible. Otherwise, these schemes are *dependent*.

Occurrence of an event indicating violence of separation standards by two aircraft using the dependent schemes S_A and S_B is strongly correlated with difference of values of the entry times $t(S_A)$ и $t(S_B)$. For example, violence can occur if time difference between $t(S_A)$ и $t(S_B)$ less 1 minute, and can not occur if time difference $t(S_A)$ и $t(S_B)$ more than 1 minute. Therefore, for any pair of dependent approach schemes, a function *Forbid(* $t(S_A/S_B)$*)* determining values of entry time differences $\Delta(B/A) = [t(S_A) - t(S_B)]$, if $t(S_A) > t(S_B)$ (if aircraft *A* enters its scheme after aircraft *B*), and $\Delta(A/B) = [t(S_B) - t(S_A)]$, if $t(S_B) > t(S_A)$ (if aircraft *B* enters its scheme after aircraft *A*) can be introduced. Detection of all dependent pairs of approach schemes S_X and S_Y for particular airport topology as well as calculation of the functions *Forbid(* $t(S_X/S_Y)$*)* (it may be represented as a matrix) may be performed in different ways including simulation–based approach. The function *Forbid(* $t(S_X/S_Y)$*)*, $S_X, S_Y \in \boldsymbol{S}$ constitutes a model of temporal constraints for simultaneous use of dependent schemes of approach zone.

The above introduced model of temporal constraints will be used as a component of air traffic control organization as follows. Current air traffic situation (configuration), at the time moment t_0 is presented by the set of aircraft Set_{InAppr} that are already operating within approach zone and their trajectories are conflict–free with regard to the safety policy. For every such aircraft of Set_{InAppr}, the approach schemes and input times (in approach zone) are known. Function *Forbid(* $t(S_X/S_Y)$*)* makes it easy to compute the earliest admissible time for every scheme after that this scheme can be used by other aircraft without conflicts with aircraft from set Set_{InAppr}.

3.2.3. Typical behavior patterns of normal aircraft in normal air traffic situations: Conceptual Description

Typical model of a normal aircraft movement intended for landing comprises the typical behavior patterns as well as negotiation acts with corresponding air traffic operator(s) as it is described below.

Entry into airport airspace

Aircraft pilot informs corresponding air traffic operator of arrival zone about the altitude and entry point of arrival zone in advance, when the aircraft is approaching to arrival zone. Depending on the situation, the pilot either receives approval of the intention and assigned arrival movement scheme or does not receive.

Behavior pattern within arrival zone

- Within arrival zone, aircraft is moving along the axes of legs constituting the assigned arrival scheme. During movement, the aircraft is passing through the arrival zone points indicating ends of the previous legs and begins of the subsequent ones.
- Passing through a scheme point
  - Every arrival scheme point is assigned the admissible altitude echelons and aircraft may pass the point using only one of these echelons that was assigned to the aircraft be the arrival zone air traffic operator.
  - In some of these points, the holding area (circles) exists. While approaching to such point, the aircraft either receives permit to entry into the subsequent leg or it is prescribed to entry to the corresponding holding area where aircraft has to wait for permission to continue movement along the next leg of the assigned scheme.
- Movement inside a leg
  - Along legs, the aircraft is moving at assigned altitude echelons while changing them during descending according to a designation.

    - If the aircraft has to outrun the other one (e.g. due to difference in admissible speeds for aircraft of different classes) both have to deviate from the leg axis at predefined distance to the different sides from. When outrun is completed the aircraft return to the leg axis and continue the movement along the latter. An important requirement is that both aircraft have to return to the leg axis before the exit point of leg.
    - The aircraft is permitted to simultaneously perform outrun evolution and echelon change evolution.
- Movement inside holding area
    - Each holding area has parameter determining the time of holding circling. Depending on the situation, aircraft may be prescribed to perform several circles within the holding area.
    - Inside holding zone, the aircraft has to move using a single altitude echelon but it is also may overcome to the lower echelon.
- Vectoring
    - Vectoring is a behavior pattern intended exit out of the leg margins. Completion of the vectoring corresponds to the coming back to a leg of the same or different arrival scheme.
    - Every vectoring evolution supposes building new trajectory of aircraft flight out of the arrival schemes and legs constituting the schemes.
    - An example of about standard vectoring evolution caused, e.g., by weather conditions, technical problems, terrorist threat, etc., is to turn at 30 degrees from the leg axis in horizontal plane, to fly 20 km and then return to the former course using, possibly, other echelon.

Movement inside approach zone

- Entry into approach zone depends on current air traffic situation and is made on permission by air traffic operator of the approach zone. Till permission, the aircraft has to wait inside a holding area of the arrival zone.
- Movement inside the approach zone is carried out along the approach scheme.
- If, due to some reason, the aircraft entered into approach zone cannot realize the landing, the latter continue its movement using a scheme of missed approach linked to its approach scheme. In this case, the aircraft returns to one of the landing trajectory. In any case, entry into new (next) landing trajectory, the aircraft needs to receive permission of the air traffic operator. Before that permission, the aircraft has to wait within a holding area specifically designated for missed approach situations.

Let us consider behavior patterns of taking–off aircraft.

Take-off

- The aircraft pilot is assigned the movement scheme and informs the air traffic operator about expected take–ff time and waits for permission for take–off. Depending on the current air traffic situation, the permission may be received with some delay.

Movement inside approach zone

- Inside this zone, the aircraft is moving according to predefined departure scheme.

Movement inside arrival zone

- Inside arrival zone, the aircraft is moving along predefined departure scheme before an exit point from airport airspace.

3.3. Organizational Structures of Air Traffic Control

3.3.1. Control functions

Previous section conceptually described rules of air traffic organization imposing constraints on admissible behavior patterns of particular normal aircraft within various zones of the airport airspace. In other words, these rules determine autonomous component of the aircraft movements. The second part of regulations concern air traffic controlling functions in different zones of airport airspace, which uniquely

determine the aircraft' movement. This part of control is controlled and triggered by some "events" that may correspond to either commands issued by air traffic operator(s) or produced in a way on board of aircraft (aircraft crew, automatic equipment, etc.). Exactly division of responsibilities between operator(s) and command crew determine what is below called "organizational structure" of air traffic control. Let us first enumerate the above mentioned controlling "command" (actually they result from solution of corresponding tasks) and then consider two organizational structures, the *existing* one and the *structure proposed* in this research. The latter is distinguished from the former by the intention to provide aircraft with more autonomy thus simplifying the responsibilities of air traffic operators.

The following types of control are currently performed by air traffic operators:

A. Permission, for an aircraft approaching to the airport airspace, to entry into the latter.
B. Permission, for an aircraft operating within arrival zone, to transit into next leg.
C. Sending directives, to an aircraft operating within arrival zone, to transit into lower altitude echelon.
D. Coordinated evolutions of aircraft operating within arrival zone, in the outrun situations.
E. Permission, for an aircraft operating within arrival zone, to entry into approach zone for the subsequent landing.
F. Changing the aircraft speed.[2]
G. Performing, by an aircraft operating within arrival or approach zone, vectoring.
H. Permission, for a taking–off aircraft, to take off.

The objectives of the air traffic control are as follows:

- Smooth on-line safe landing and taking off the aircraft assumed by timetable of their arrivals and departure
- Providing safety of aircraft operating within airport airspace via support of separation standards
- On-line optimization of the air traffic minimizing total delays of aircraft.[3]

Although the analysis of air traffic control organizational structures given below concerns normal situations, in many respects it may be extended also to the abnormal situations when, e.g., a hijacked aircraft arrears in the airport airspace. In the latter case the air traffic model of normal aircraft remains unchanged but decisions listed above in items A)–H) are made in more constrained situations determined by new threats. That is why the role of autonomy of aircraft in safety provision is increased and more control function should be performed automatically by the software agents assisting the pilots and cooperating with each other via P2P negotiations.

3.3.2. Existing and Proposed Organizational Structure of Air Traffic Control

The air traffic control organizational structure that is currently in use is illustrated in Fig. 3.11.

The main participants of the existing organization are as follows:

- *Aircraft' crews (*pilots of the aircraft) and
- *Air traffic operators* responsible for some control functions in various sectors of the airport airspace.

According to the existing organizational structure of air traffic control, all decisions enumerated above in the items A,..., H are produced by air traffic operators of the corresponding sectors.
Accordingly, two main roles of the air traffic control domain organizational structures are "*pilot*" and "*air traffic operator*". The latter notice is important due to the fact, that multi-agent paradigm of and "*role–based*" methodology for multi-agent software development is below used.

The air traffic control organizational structure that is proposed in this Project and used below is illustrated in Fig. 3.12. The main participants of this organization are as follows:

- *Aircraft' crews (*pilots of the aircraft) and
- *Air traffic operator of approach zone which is responsible for making decisions listed above in items* E, F, G, H concerning aircraft' movement within approach zone.

---

[2] Commands marked as f) and g) are not planned for implementation within this Project. This is a simplification assumed.
[3] This issue is not so far considered in the Project.

- Thus, aircraft' crews, in the proposed structure, are responsible for autonomous solutions on the tasks A, B, C, D, F, G. Two important issues constitute the basis of control functions of the aircraft' crews: 1) organization of information exchange and 2) safety policy determining the aircraft's autonomous behavior. Let us consider these issues.

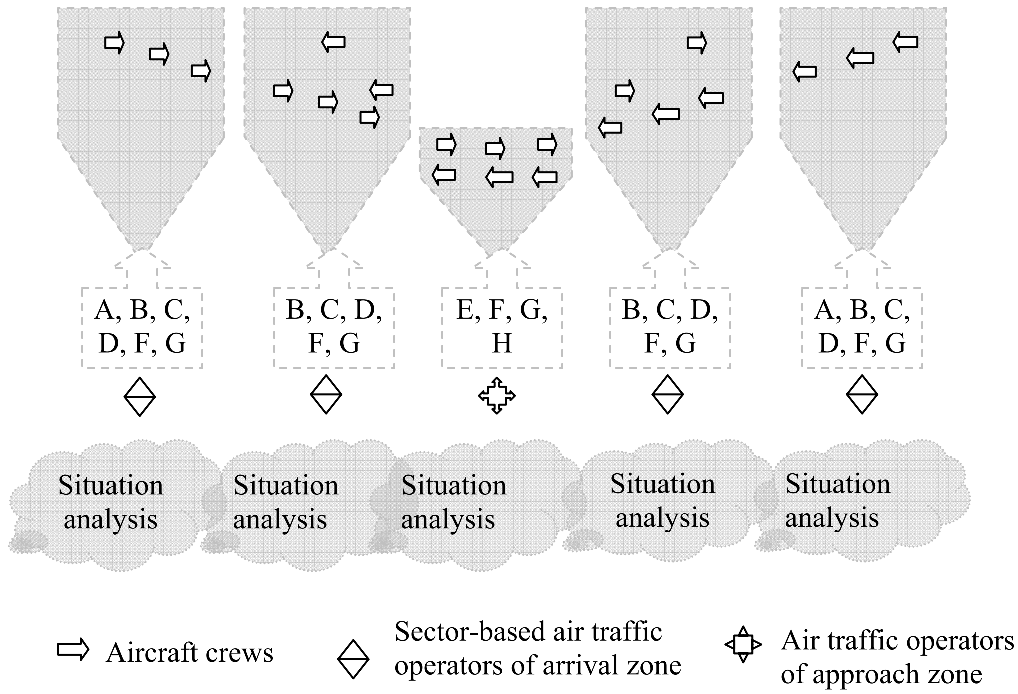


Fig. 3.11. Existing organizational structure of air traffic control within airport

3.4. Organization of information exchange

Autonomous behavior in constraints environment, in airport airspace, assumes that each aircraft has to possess the information about state of the environment that is constituted by other moving aircraft. That is why each aircraft has to possess information about state, speed and course of other aircraft and their anticipated movement plans. The simplest approach is to organize broadcasting when each aircraft informs the other ones about its movement attributes and future plans. Unfortunately, this may lead to too high communication overhead and, therefore, delays in decision making. On the other side, if a pair of aircraft uses non–overlapping arrival schemes then they do not need to know above information concerning each other that is a consequence of organizational principles associated with the safety issue described in subsection 3.2.2, because no unsafe movement leading potentially to the violation of the separation standards may occur for this pair.

This fact may be used for *decomposition of the aircraft of arrival zone in independent groups* such that only aircraft of the same group need to interact to prevent conflicts while aircraft belonging to different groups do not need to interact. Group formation and, therefore, *decomposition* may to be done *on the sector basis*, where the sectors are determined as the component of airport airspace topology as follows:

- The whole approach zone is a sector.
- Arrival zone is divided into sectors in the following way. The total count of the sectors is determined by the total count of points within arrival zone in that holding areas are determined. It is convenient to identify any sector by the name of the respective point. In addition to the holding area, each sector contains several legs which are determined in the way indicated below. Let *id* is identifier of entry point of a holding area. Then sector *id* is composed of the sequence(s) of legs belonging to one or several arrival schemes with the following properties: (a) sequence of the legs is ended in the point *id*, and (b) sequence of the leg starts just in the other point of the arrival scheme where the previous holding area is located.

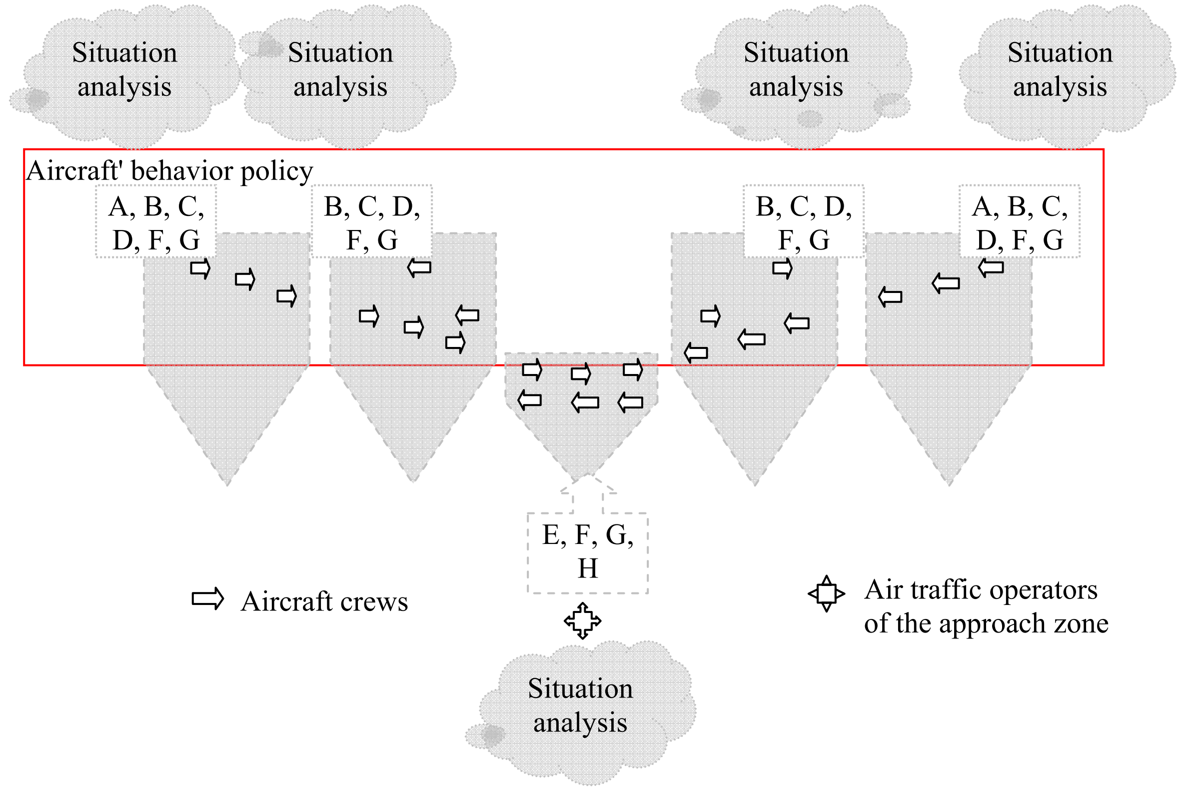


Fig. 3.12. Proposed organizational structure of air traffic control within airport

As a result of such splitting, the set of aircraft operating in the arrival zone is decomposed on several overlapping groups, and each sector *Sector* is assigned one of such groups, *group*(*Sector*), which is denoted by the same identifier *id* as corresponding sector. Since arrival scheme may include several sectors, each aircraft may belong to several groups depending on its behavior strategy and current air traffic situation.
One of two basic principles may be used for determining aircraft membership to this or that group:

Principle 1. At any time, when an aircraft is located within arrival zone, it belongs to all groups of the set {*id*} where {*id*} corresponds to the set of identifiers of all the sectors belonging to the assigned arrival scheme.

Principle 2. At any current time *t*, an aircraft belongs to only two groups corresponding to the sector of its current location id_1 and to the next one, id_2, of its arrival scheme in which it will transit.

Of course, intermediate variants are also possible. Final selection of a grouping principle is the subject of software prototyping–based experimental research planned for the second phase of the Project. Nevertheless, some properties of both variant may be formulated speculatively, i.e. without experiments:

- The information exchange needed to compute conflict–free behavior of aircraft in arrival zone is held in both cases.
- Potentially, due to more complete information provided for particular aircraft, the better quality of air traffic control can be provided, but it leads to more intensive information exchange and therefore, to greater communication overhead that may contain unnecessary messages.

According to the proposed air traffic control organizational structure, the aircraft have to autonomously solve the tasks A, B, C, D, F, G described above. Analysis of the information needed to the aircraft in order to autonomously cope with these task solutions shows that aircraft have to exchange information presented in Tab. 3.1. Let us note that each aircraft have to possess the indicated information concerning all aircraft of groups to which it belongs at current time instant.

Table 3.1. Information to be on-line exchanged by group aircraft

Aircraft's data	
Aircraft	*<Aircraft's identifier>*
Class	*<Aircraft's class>*
Current sector	*<id of sector in which aircraft is currently located>* ,
Next sector	*<id of sector into which the aircraft has to overcome next>*
Update time	<t*ime of information update*>
Movement related data	
On Altitude	*<Current altitude echelon>*
To Altitude	*<Next selected altitude echelon>*
In holding area	<Holding area usage>
Information related to transit into next sector maneuver	
Transition point	*<Name of entry point>*
Transition time	*<Next sector transition time>*
Transition status	*<Intention /Decision>*
Approach	<Flight Scheme within the next zone> (Only for the aircraft of the approach zone)
Schedule deviation	
S-Delay	*<Accumulated delay>*
F-Delay	*<Total accumulated delay of the flight>*

The important notices concerning the above given data are as follows:

Notice 1. Data are updated if only aircraft produces a decision of the set {A, B, C, D, F, G}. In other words, if aircraft produces some of the aforementioned decisions it obliges to refine the attribute values in order to inform all other group member aircraft about its decision.

Notice 2. While receiving the updated information the aircraft's software has to evaluate their influence on its own admissible movement, in particular, from safety viewpoint and potential solutions for its tasks A, B, C, D, F, or G. The most important data are the data related to the current aircraft's position (coordinates) at current time instant. It is clear that position–related data are needed for evaluation of the aircraft of the group current and future safety. When necessary, an aircraft may to additionally request information concerning group–based information exchange.

3.5. Typical Behavior Patterns of Normal Aircraft: Simplifications and Formal Specification

It is assumed that, when an aircraft enters into arrival zone, the latter autonomously constructs the movement plan for the current and forthcoming sectors of the assigned arrival scheme. This planning is quasi–local and concerns at most two adjacent sectors. When necessary they re-compute their movement plans to achieve conflict free movements. At that, new plan (1) covers the aircraft movement to the end point of the current or next sector and (2) supposes achievement of the above mentioned point provided that its height corresponds to a free echelon. It is important to note that using of the holding zone attached to the end point of the sector is assumed in two cases: (1) the aircraft did not find a conflict–free continuation of its flight without use of holding zone and has to find it with some delay corresponding its movement within holding zone or (2) use of the holding zone is a behavior pattern of its plan. The time instants when an aircraft re-computes its plan are explained graphically in Fig. 3.13.

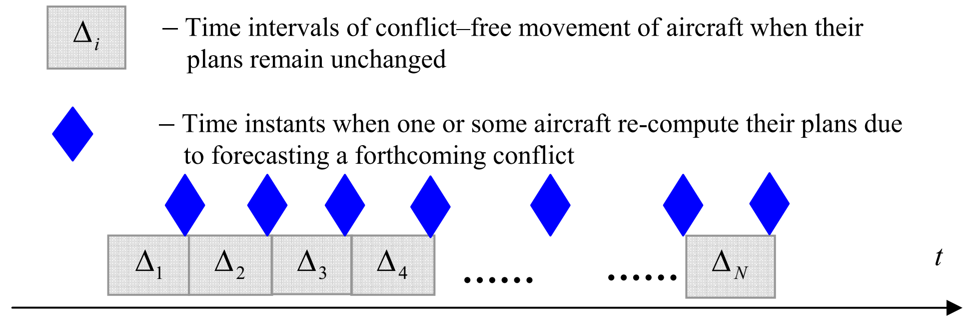


Fig. 3.13. Time instants when an aircraft re-computes its plan

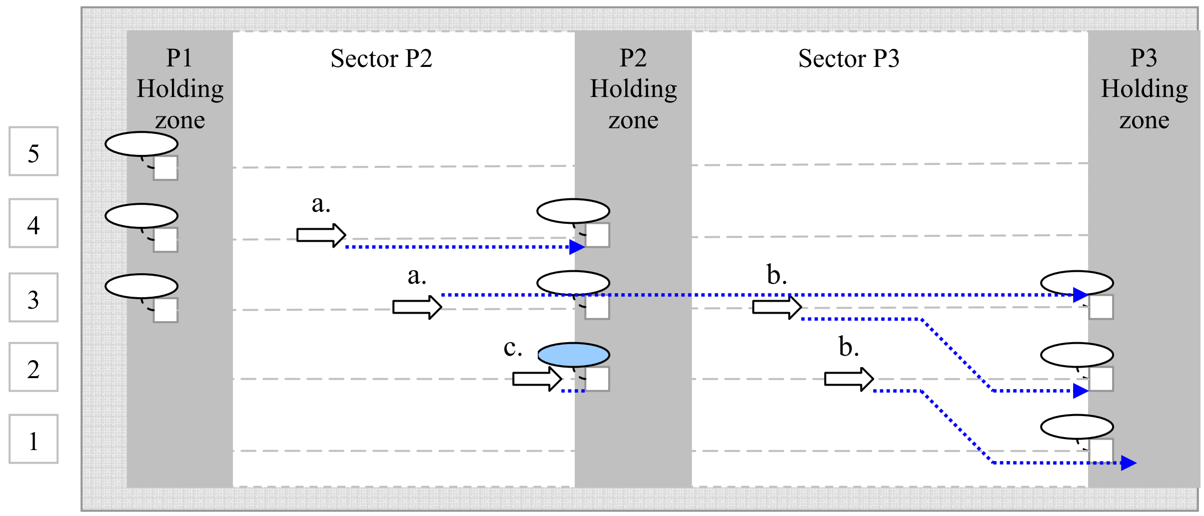


Fig. 3.14. Behavior patterns of normal aircraft

Let us note that a plan is ordered sequence of behavior patterns in which the starting point of any pattern coincides with the end point of the previous one.

The following behavior patterns[4] of normal aircraft may be considered as typical ones::

a. Horizontal movement within given echelon up to a given point along the leg axis line[5];
b. Change of the current echelon while moving along the leg axis line up to the achievement of the destination echelon;
c. Movement within the holding zone in given echelon.

The behavior patterns *a*, *b* and *c* are used in the air traffic model and deconfliction algorithm developed during the first phase of the research. Of course, this list is not exhaustive one. Behavior pattern listed below also are in used of normal aircraft, but they are not used in current model of normal aircraft movement and are intended to be used in the next phase. Of course, this is a simplification assumed in current version.

d. Change of echelon while moving within the holding zone;
e. Movement inside the leg zones within given echelon. Let us note that this pattern does not assume movement along the leg axis line;
f. Change of echelon while moving inside the leg zones up to achievement of the given point of the destination echelon;
g. Vectoring within given echelon up to achievement of the given point;

---

[4] Each behavior pattern except one corresponding to a holding zone is linear segment of trajectory through which the aircraft is moving evenly.
[5] Axis line is the projection of the vertical plane formed by medial points of a sector leg.

h. Vectoring with echelon change echelon up to achievement of the given point.

Formal specification of the aforementioned behavior pattern is done in terms of the attributes listed in the Tab. 3.2. Their mapping to the attributes of typical behavior patterns of normal aircraft is given in Tab. 3.2.

Table 3.2. Attributes used for formal specification of the typical behavior patterns of normal aircraft.

Basic specification attributes		
Point	(x, y)	Current coordinates of an aircraft or a point
Altitude	*H*	Echelon
Attributes to be computed		
Time (*Point*)		Time instant of the destination of the point *Point*
Velocity	*V*	Current horizontal velocity
	dV	Acceleration / deceleration in horizontal plane
	dH	Vertical velocity
Direction	*0..360*	Course to point *Point*
Type	*0 .. 3*	0 – belongs to the leg *Leg* ; 1 – end point of the leg *Leg* ; 2 – belongs to the inside area of the leg *Leg*, 3 - – located outside of the airport airspace legs
Leg	*0 .. 2*	0 – movement along the axis line of the leg *Leg* ; 1 – movement inside the leg *Leg*; 2 – movement outside of the airport airspace legs
Air Space	S_1& S_2	Identification of the arrival schemes in between which the considered part of airspace is located

Table 3.3. Mapping attributes to the typical behavior patterns of normal aircraft

Attributes	*Behavior patterns*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*
Point	*(X, Y)*	*P*	*P*	*P*	*P*	*P*	*P*	*P*	*P*
Altitude	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
Time		*t*	*t*	*t*	*t*	*t*	*t*	*t*	*t*
Velocity	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
	dV			*dV*	*dV*		*dV*		*dV*
	dH			*dH*	*dH*		*dH*		*dH*
Direction	*0..360*	*D*			*D*	*D*	*D*	*D*	*D*
Type	*0 .. 3*	*0, 1*	*1*	*1*	*0, 1, 2*	*0, 1*	*0, 1, 2*	*0, 1*	*0, 1*
Leg	*0 .. 2*	*0*			*0*	*1*	*1*	*2*	*2*
Air Space	S_1& S_2							S_1& S_2	S_1& S_2

Let us note that any aircraft has a movement plan up to the given point, but after it, it is assigned only an arrival scheme in terms of legs.

The above described specification of the normal aircraft behavior patterns determines formally the admissible collective (mutual) movements of the normal aircraft which also, for safety purposes, have to meet the safety policy. Checking and support of air traffic safety policy is strongly based on admissible behavior of the aircraft subject to separation standards. Safety policy determines the classes of control commands (see section 3.3.1) and events causing the necessity of coordinated re-planning intended to

meet separation standards. The safety policy rules ("social rules" in terms of notions used in multi-agent framework) are intended to predict and avoid usage, by any pair of aircraft, conflicting plans while minimizing the negotiation overhead.

3.6. Typical Behavior Patterns of Hijacked Aircraft

Practically, any behavior pattern of normal aircraft may be also used by hijacked one. Moreover, if to use vectoring as a behavior pattern of normal aircraft the set of behavior patterns of both, normal and hijacked, aircraft may use the same set of behavior patterns. An important difference between motion of normal and hijacked aircraft is that it may ignore commands of the air traffic operator and not follow the rules of air traffic organization within airport airspace (using predefined legs, waiting zones, entry and exit points of airport airspace, may violate predefined height echelons, etc.). There may be one or several hijacked aircraft, but in this research we consider only the case of sole hijacked aircraft.

An exhaustive list of types of behavior patterns of a hijacked aircraft cannot be anticipated. In this research several variants of such patterns presenting various degrees of threat for normal aircraft, are considered. Let us briefly describe them.

We have distinct the behavior patterns according to several attributes:

1. Area of airspace when the fact of hijacking became known for air traffic operator: *out of the airport airspace*, *within the arrival zone*, *within the approaching zone*. One more variant is if the hijacking occurs in the airfield. The latter case is not considered in the Project because it is not "in flight" case.

2. Target of hijacked aircraft motion: *departure* of the airport airspace, *landing*, *flight* within airport airspace without definitely clamed target or only *crossing* it.

3. Type of trajectory: *translation* (with constant speed and course, horizontally or ascending/descending), "broken line" when aircraft changes, from time to time, course and speed; movement along a circle (which may be not mandatory a holding area).

In this Report we do not pay a lot of attention to variety of behavior pattern. The reason is that thorough analysis of the developed deconfliction algorithm, its deep verification and complexity analysis is the task of the second phase of the research when multi-agent implementation of the airspace deconfliction system is done. At current stage of the research, only several variants of typical behavior patterns will be used for verification of the basic airspace deconfliction algorithm. They are (a) patterns corresponding to translation of hijacked aircraft within arrival zone; (b) using "broken line" trajectory. In both cases, the case of *flight* within airport airspace without definitely clamed target will be basic ones. In some sense, the aforementioned cases correspond to the heavy enough situations and that is why they can be used for preliminary verification and assessment of the airspace deconfliction algorithm if the latter is implemented in non-multi-agent environment.

As concerns formal specification of the hijacked aircraft behavior patterns, the same kinds of patterns as for normal one will be used, i.e. behavior patterns *a*, *b* and *c* described in section 3.5. Of course, these behavior patterns of hijacked aircraft determine some kind of simplification of the deconfliction problem statement, but it looks admissible at current research phase where the main efforts are associated with development the design project of the multi-agent airspace deconfliction system and development and verification of the basic deconfliction algorithm. However, the main distinctions of these patterns for hijacked aircraft in comparison with normal ones are that the former take place out of the standard elements of airport airspace topology.

On the basis of the aforementioned behavior patterns a quite complex scenarios of hijacked aircraft can be constructed. These scenarios represent behavior of hijacked aircraft in context of normal air traffic. More precisely, these scenarios describe behavior of hijacked aircraft as potential threat for various groups of aircraft.

An important issue concerning deconfliction problem is classification of the types of threats born by hijacked aircraft with regard to normal aircraft. The latter is important due to the fact that such classification used in deconfliction algorithm may lead to a decomposition of the deconfliction task. Below the following types of threats are considered:

$G_1(t, X, HA_j)$ – a group of normal aircraft belonging to sector *X*, for which hijacked aircraft HA_j is conflicting at current time instant t;

$G_2(t, X, HA_j)$ – a group of normal aircraft belonging to sector *X*, for which hijacked aircraft HA_j does not present a threat at current time instant t, but the conflict will mandatory occur if the normal aircraft will keep their movement plans unchanged, and hijacked aircraft will be moving according to its predicted trajectory;

$G_3(t, X, HA_j)$ – a group of normal aircraft belonging to sector *X*, which have conflict–free movement with regard to hijacked aircraft HA_j at current time instant t and later if the normal aircraft will keep their movement plans unchanged, and hijacked aircraft will be moving according to its predicted trajectory,. However, there exists a maneuver of the hijacked aircraft which will result in a conflict with normal aircraft.

3.7. Chapter concluding comments

This Chapter presented the developed realistic conceptual model of the safe air that has to provide the aircraft motion within given boundaries and meeting the separation requirements given in terms of minimum allowable distance between aircraft. This result corresponds to the solution of the Task 2 of the Work plan. It also describes typical behavior patterns of normal and hijacked aircraft and air traffic configurations to be modeled in the Project as well as organizational structures of air traffic control that are assumed by the Task 3.

4. Airspace Deconfliction Algorithm

Airspace deconfliction algorithm is described below together with the simulation environment intended for simulation–based verification of the former that is assumed by Work Plan. Simulation environment is a software tool that provides user with friendly interface needed to simply support for specification of the dynamic situations within airport airspace, i.e. time–dependent behavior of normal and hijacked aircraft operating in common space when normal aircraft strive to provide safety using deconfliction algorithm. Below this deconfliction algorithm is described in a centralized form, because, at current phase of the Project research, only algorithm itself and verification of this algorithm has to be done. Development of its distributed form when deconfliction task is solved using distributed algorithm represented as a set of local safety policies is the subject of the next research phase. It will be implemented as P2P negotiations of autonomous agents assisting the pilots.

The basic idea of the developed deconfliction algorithm is as follows. To decrease computational complexity of the algorithm, it is organized in two steps. At the first step, all aircraft operating within airport airspace that potentially may conflict with hijacked aircraft are ordered according to their priorities. In fact, priorities determine the order in which the aircraft will be permitted to use "resource" that is the airport space (legs, sectors, holding areas, runways, etc.). Then aircraft, in predefined order, autonomously plan their movements using the resources of airport space that remained free or became free again.

Let us note that an assumption used in current phase of development of airspace deconfliction algorithm is that, in it, along with hijacked aircraft, only normal aircraft entering into airport airspace and intending for landing are taken into account. Taking of aircraft are so far ignored in the considered air traffic model and corresponding deconfliction algorithm.

4.1. Conceptual Description of the Deconfliction Situations, Deconfliction Scenario and Simulation Cycle

Each normal aircraft *Y* entering into airport airspace is initially specified by a set of the following attributes:

1. $T_{Entry}(Y)$ – time instance of the aircraft entry into airport airspace;
2. *Ep(Y)* – entry point of aircraft which is represented by its name (particular *id*) and coordinates;
3. *Ranway_id* – name of airport destination runway;
4. Class of aircraft.

The subsequent movement of normal aircraft is determined by its plan of movement in arrival zone formed autonomously in real-time mode and its movement within approach zone where air traffic operator prescribes the aircraft the landing plan.

As concerns hijacked aircraft, its movement is specified as a sequence of its behavior patterns (see section 3.5) on time scale. Each behavior pattern is determined by coordinates of its initial and end points and time instances assigned to the former and the latter. Current position, speed and course of the hijacked aircraft may be computed if to assume that its speed, in between of two aforementioned points, is even and, therefore, its intermediate positions including current one may be interpolated. Of course, for external "observers" of the hijacked aircraft only current position, speed and course are available. Future trajectory of the hijacked aircraft may be only predicted with some error while assuming some hypothesis concerning its future movement.

Situation within airport airspace at a time instant *t* is understood as a set of normal aircraft assigned current position, speed and course together with the name of target runway and current position, speed and course of the hijacked aircraft.

Simulation of the current situation and its development in time is provided by simulation environment. Airspace deconfliction algorithm is simulated within simulation environment, within which the developed software has to be considered as a changeable component of simulation environment. High-level structure of the simulation environment without user interface part is presented in Fig. 4.1. Simulation is organized on discrete time basis. This means that some external components generates input events evenly in time,

$t = t_0$, $t_0 + \Delta$, $t_0 + 2\Delta$, …, which intervals are considered as even simulation cycles. Fig 4.1 presents deconfliction scenario and an arbitrary simulation cycle execution scheme.

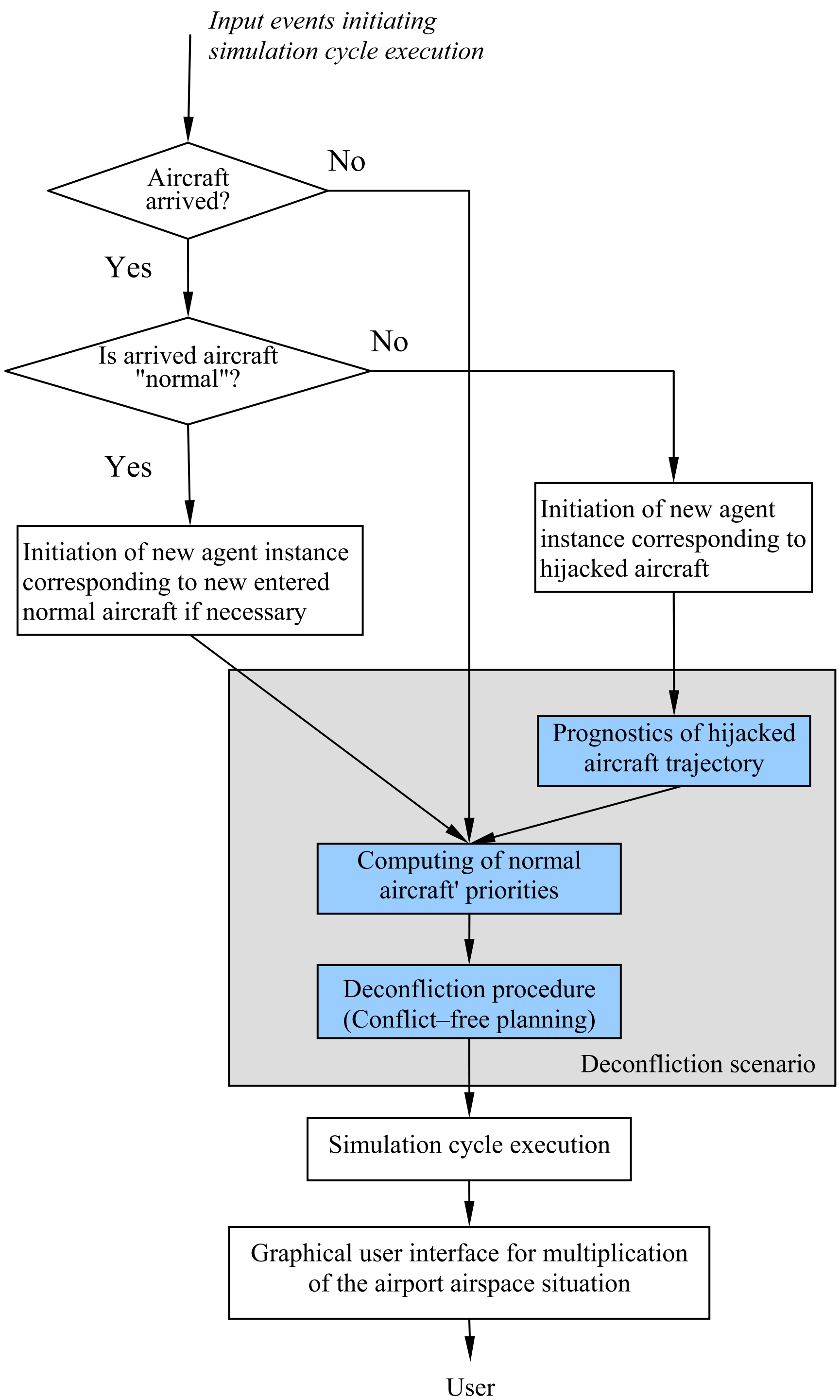


Fig. 4.1. Deconfliction scenario and situation simulation environment

Let us explain deconfliction scenario and situation simulation environment given in Fig. 4.1

Input events initiating simulation cycle execution

This is simple discrete time-based event generator intended to launch the next simulation cycle at the next discrete time instant.

Checking of entrance of new aircraft into airport airspace

This procedure checks the airport arrival timetable to detect whether new aircraft will arrive during the forthcoming time interval (simulation cycle).

Checking appearance of hijacked aircraft

Time and attributes of the hijacked aircraft are recoded in a data base and this procedure only checks whether the subsequent hijacked aircraft fall into the time interval corresponding to the forthcoming simulation cycle.

Initiation of new agent instance corresponding to new entered aircraft if necessary

If previous procedure detects new aircraft arrived into airport airspace it read its attributes from data base and generates new instance of pilot assisting agent with the attributes of new aircraft.

Checking appearance of hijacked aircraft

This procedure is about the same as for normal aircraft with the sole difference that appearance time of the hijacked aircraft is manually recorded in data base by user.

Initiation of new agent instance corresponding to hijacked aircraft

This procedure is about the same as for normal aircraft with the sole difference that attributes of the hijacked aircraft are manually recorded in data base by user.

Forecasting of hijacked aircraft trajectory

Forecasting of the hijacked aircraft trajectory is done using assumption that it is moving evenly while preserving constant speed and course. Such forecast is assumed to be true up to the subsequent time of aircraft's maneuver recorder in data base. The core objective of this procedure is to detect current and future potential conflicts of the hijacked aircraft with the normal ones. Conflicts are computed using separation standards introduced for the case of hijacking and forecasted trajectories of hijacked aircraft and plans of normal aircraft.

Computing of normal aircraft' priorities

This procedure determines priorities of the normal aircraft to re-plan their future movements. Since each aircraft independently computes its own plan in context of plans of other aircraft, the order in which local plans are computed influence on the quality of the resulting plan. Priorities determine the order of planning for various aircraft. Priorities are computed using expert-based rules.

If during the forthcoming time interval an aircraft is ready to entry into approach zone with the subsequent landing it asks permission for this entry from air traffic operator. If the permission is received the aircraft is assigned the plan of movement within approach zone. Otherwise it uses the sector's holding area to wait the permission.

Deconfliction procedure (Conflict–free planning)

Actually, this is the core of deconfliction procedure. This procedure is used by normal aircraft for conflict–free planning of their subsequent movements. This algorithm will be described in detail below. Using this algorithm, every aircraft either refines its plan of movement within current sector or re-plans its movement within the next one.

Simulation cycle execution

This procedure computes the attributes of normal and hijacked aircraft' attributes (position, speed, course) corresponding to the right point of the simulation cycle.

4.2. Air Traffic Situation Prediction

The main objective of this procedure is to detect existing and future conflicts determined by the presence of hijacked aircraft. This procedure compute current and forecasts future positions of the hijacked aircraft at the time instants $t \in \{t_c,\ t_c+\Delta, ..., t_c+n\Delta\}$, where t_c is current simulation time and n is determined by the selected forecast interval.

For every time instant of the forecast interval, the minimal distances between hijacked aircraft and each leg of the sectors, $D_{\min}(t, Leg)$, is computed. Fig. 4.2 explains how these distances are computed.

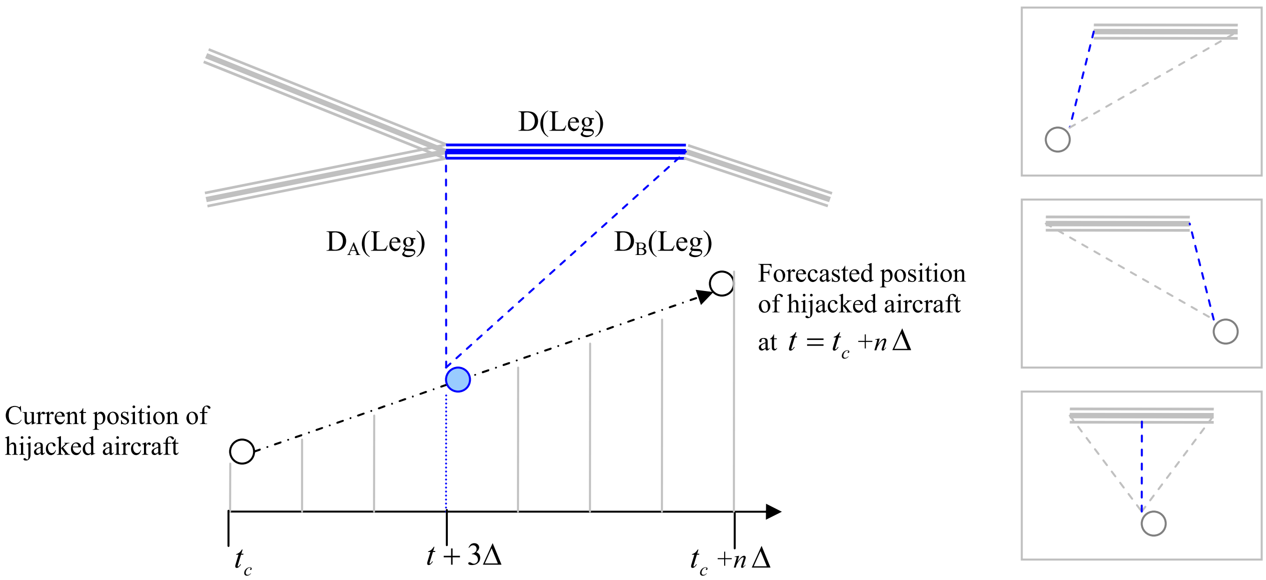


Fig.4.2. Forecasting the movement of hijacked aircraft and evaluation of minimal distance between the former and sector legs

If $\{Leg_1,\ Leg_2, \ldots, Leg_r\}$ is the set of legs of a sector *Sector* and $\{D_{\min}(t, Leg_1),\ D_{\min}(t, Leg_2),\ D_{\min}(t, Leg_r)\}$ is the set of minimal distances between hijacked aircraft and aforementioned legs respectively then

$$D_{\min}(t, Sector) = \min\{D_{\min}(t, Leg_1),\ D_{\min}(t, Leg_2)\ D_{\min}(t, Leg_r)\}$$

Every sector *Sector* is then assigned a value of function $Conf(t, Sector)$ qualitatively evaluating potential threats caused by hijacked aircraft. This value is computed according to the following rule:

$$Conf(t, Sector) = \begin{cases} 1, & \text{if } D_{\min}(t, Sector) > D_1 \\ 2, & \text{if } D_1 > D_{\min}(t, Sector) > D_2 \\ 3, & \text{if } D_2 > D_{\min}(t, Sector) \end{cases}, \tag{1}$$

where values D_1 and D_2 are determined by separation standards. Conceptually, the equation (1) may be interpreted as follows:

If any normal aircraft is moving inside the sector *Sector* and value of qualitative function

- $Conf(t, Sector) = 1$ then the former will not conflict with the hijacked aircraft at time instant *t;*
- $Conf(t, Sector) = 2$ then the former will conflict with the hijacked aircraft at time instant t if normal aircraft moves according its current plan and hijacked aircraft undertakes some maneuver;
- $Conf(t, Sector) = 3$ then the former can conflict with the hijacked aircraft at time instant t if hijacked aircraft moves according to its forecasted trajectory;

Fig. 4.3 explains how the function $Conf(t, Sector)$ is computed.

4.3. Priorities and Ordering of Normal Aircraft in Deconfliction Procedure

Priorities are used to order the aircraft thus determining the order in which the aircraft solve their deconfliction tasks. Such class of tasks is well known within general resource constraint resource allocation and scheduling and usually it is solved using so called "priority rules" that are either extracted from domain experts, or result from an example–based machine learning procedure. In this research, priority rules are designed using domain expert-based approach.

General idea of normal aircraft ordering is as follows. First, for particular airport, partial order over the sectors of airport airspace is introduced. This order is determined for any pair of the airspace sectors as "geometrical" precedence. Indeed, the set of sectors of airport airspace is ordered in sequences leading from entry points to runways. At that, within arrival zone, when an aircraft has entered through particular entry point its trajectory further follows along the uniquely predefined sequence of the sectors that are used by this aircraft during its flight from entry point till last sector of the arrival zone. Thus, within

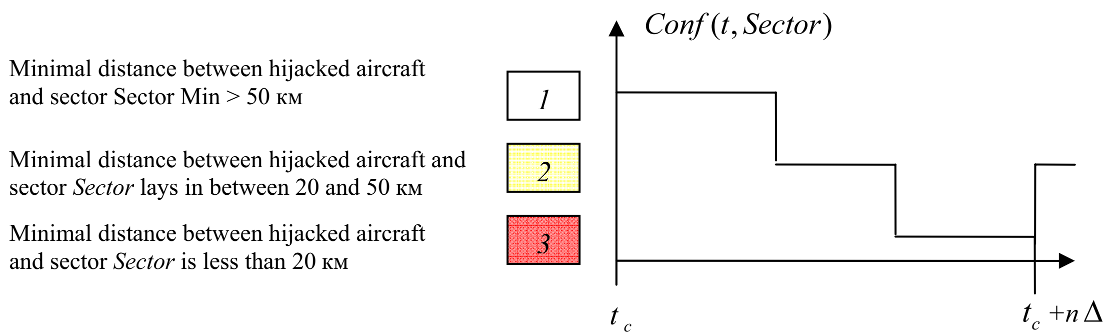


Fig. 4.3. Qualitative evaluation of conflict emergency between normal aircraft operating within sector *Sector* and hijacked aircraft

arrival zone, the sectors' structure is ordered as hierarchy that is a type of partial order which can be presented as immediate precedence relation. Formal definition of this precedence relation is as follows.

Definition. It is said that sector X_i immediately precedes the sector X_j $(X_i < X_j)$ if the former is the next sector in the landing trajectory of an aircraft.

Using such relation makes it possible to organize the aircraft ordering procedure in two steps. At the first step, the aircraft' groups formed on sector basis[6] are ordered according to the order of group sectors, i.e. Group(*sector* X_i) immediately precedes Group(*sector* X_j) if $X_i < X_j$. Therefore, groups become ordered in the same way as its attributes, sectors.

At the next step, the aircraft belonging to the same group are ordered according to some set of rules (see below).

Let us outline the basic rules used in deconfliction procedure.

<u>*Rule 1*</u>

Let $Y_1 \in group(X_1)$ and $Y_2 \in group(X_2)$ and $X_1 < X_2$
then aircraft Y_1 has higher priority than aircraft Y_2

<u>*Rule 2*</u>

If sector X_i is the sector in arrival zone and both aircraft, Y_1 and Y_2 belong to the sectors that immediately precede the sector X_i then their priorities are determined either by *Rule 3* if no hijacked aircraft exists in airport airspace) or *by Rule 4* (in presence of hijacked aircraft).

[6] See descriptive definition of the set of aircraft *group*(*Sector*) in subsection 3.4.2.

Rule 3

Let two aircraft Y_1 and Y_2 have planned times of their exit from the current sector (sectors) $t_{Sector\ Exit}(Y_1, t_c)$ and $t_{Sector\ Exit}(Y_2, t_c)$ respectively. If $t_{Sector\ Exit}(Y_1) < t_{Sector\ Exit}(Y_2)$ then priority of Y_1 is higher than priority of Y_2.

Let us note that in general case more attributes have to be taken into account in the process of ordering the aircraft of the same sector, for example,

- Class of aircraft (it determines the diapason of speed of aircraft depending on its height); К
- Current echelon occupied by aircraft; Занимаемый эшелон высоты,
- Current difference of the aircraft's attributes from the scheduled ones;
- Fuel rest,
- Etc.

These rules require more expert-based knowledge as well as more experimental data. Some of such rules can be created at the next phase of the research if additional information is available.

Let us note that the aircraft may belong at the same time to different sectors but intend to use the same sector as next one. In this case the *rule 3* is also applicable.

If two aircraft occupy different sectors but potentially may to conflict with hijacked aircraft the functions $Conf(t, SectorX_1)$ and $Conf(t, SectorX_2)$ computed for corresponding sectors $SectorX_1$ and $SectorX_2$ have to be taken into account.

Rule 4

If normal aircraft $Y_1 \in SectorX_1$ and $Y_2 \in SectorX_2$, and $Conf(t, SectorX_1) > Conf(t, SectorX_2)$, then priority of Y_1 is higher than priority of Y_2.

4.4. Normal Aircraft Movement Planning

Planning of normal aircraft movement uses analysis of potential conflicts with hijacked aircraft in two sectors: current and subsequent ones, $SectorX_i$ and $SectorX_{i+1}$. Potential conflicts are evaluated in the following way. Aircraft computes time interval when it will be within these sectors and uses values $Conf(t, SectorX_i)$ and $Conf(t, SectorX_{i+1})$ of these sectors during computed time intervals. From conceptual point of view, the planning procedure is based on the followings rules.

If value $Conf(t, SectorX_{i+1})=1$ then airplane uses usual procedure of planning.

If value $Conf(t, SectorX_{i+1})=2$ then airplane uses procedure of planning accounting some additional constraints.

If value $Conf(t, SectorX_{i+1})=3$ and $Conf(t, SectorX_i)<3$ then transition to sector $SectorX_{i+1}$ is prohibited.

If value $Conf(t, SectorX_i)=3$ then if it is necessary airplane has to re-compute its movement plan accounting value $Conf(t, SectorX_{i+1})$

Structural diagram of planning algorithm is depicted in Fig. 4.4.

Let us explain the algorithm presented in Fig. 4.4.

Aircraft is in approach sector

If normal aircraft is in approach sector then it already has plan of movement in this sector up to landing. According to taken assumption plan of movement within approach sector is not changed.

Aircraft is in arrival sector for which the Conf( ) function value is equal to 3

In this case the aircraft agent has to check existence of potential conflict.

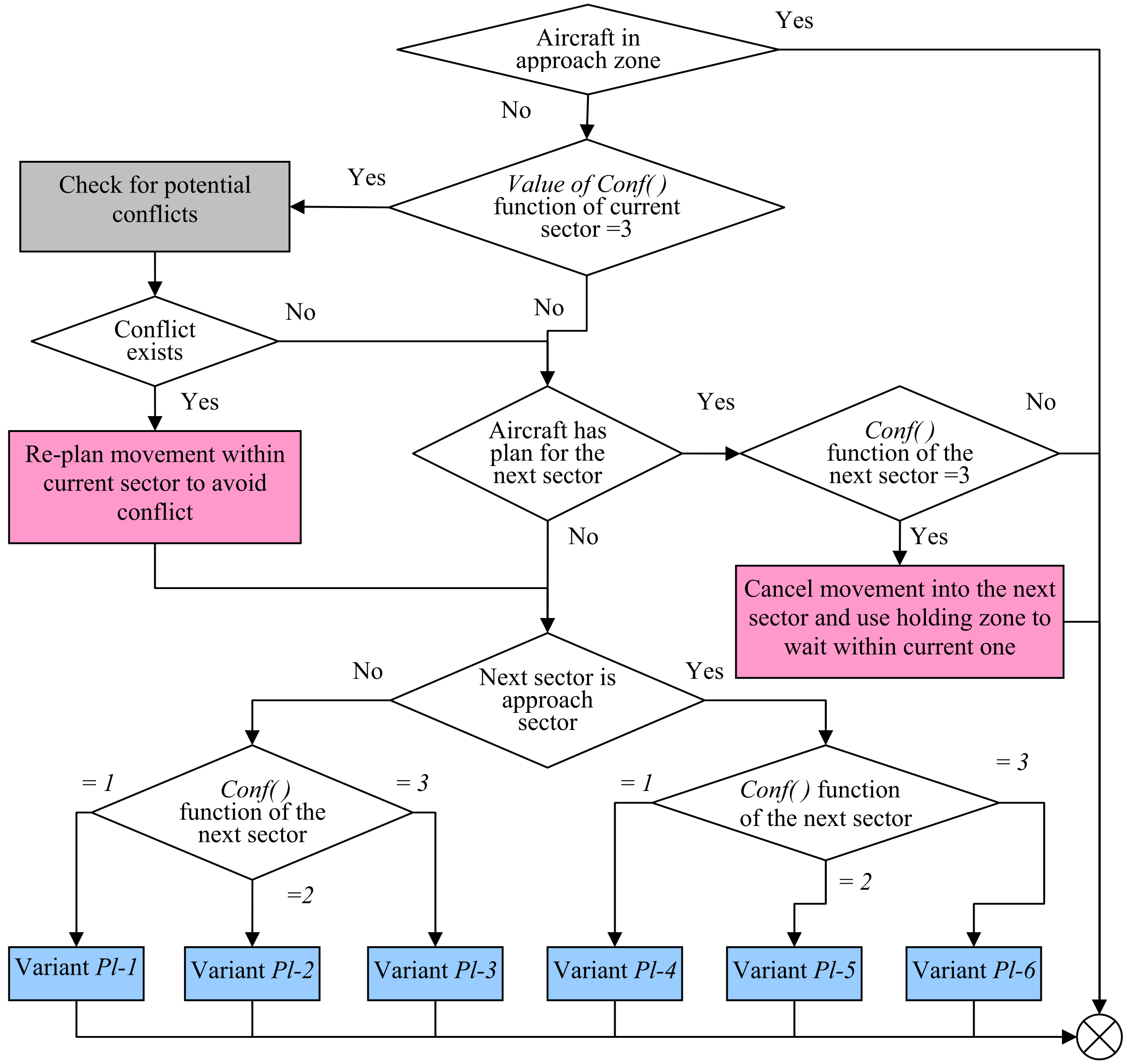


Fig. 4.4. Normal aircraft planning algorithm

Existence of conflict checking

If airplane is in the sector with value 3 it does not mean that it already has conflict with hijacked airplane. To check the latter the aircraft has to forecast the trajectory of the hijacked aircraft.

Conflict exists

If conflict is forecasted then aircraft has to re-plan its trajectory within current sector.

Re planning trajectory within current sector

In this case the basic solution is to change the echelon providing meeting of separation standard applicable with regard to hijacked aircraft. To use this solution, several other conditions have to be checked, in particular

- ✓ Existence of free echelon providing conflict–free condition;
- ✓ Existence of the transient trajectory that meets separation standards with regard to other normal aircraft;
- ✓ Admissibility to make this decision in autonomous way, i.e. without negotiation with other normal aircraft.

It needs to note that these conditions have to be provided. It is one of the main goals of proposed planning procedure in whole.
If at least one of the aforementioned conditions is not held then re-planning has to either (1) based on negotiation with other aircraft or (2) assume using of vectoring maneuver.

Checking the Conf( ) function for the next sector and *Cancel the plan developed for the next sector and remain in the current one (using sector's holding area)*

If aircraft has plan for the next sector then it has to check value of function $Conf\ (t, SectorX_{i+1})$. If this value is equal to *1* or *2* the previous plan is remained unchanged, otherwise entrance to the corresponding sector is prohibited. Otherwise, if value is equal to 3, aircraft has to cancel this plan and makes decision to occupy the sector's holding area with possible change of the echelon.

Aircraft has plan for the next sector and *Checking the Conf( ) function for the next sector*

If aircraft has no plan for the next sector then it choices the procedure of planning depending on the two conditions: (1) is the next sector approaching one or not and (2) what is the value of the *Conf( )* function for the current sector.

Let us consider possible variants of planning.

Variant Pl-1. Next sector belongs to the arrival zone and Conf(Next sector)=1.

This situation corresponds to the cases when hijacked aircraft either is absent or situated too far to violate the separation standards with regard to the normal aircraft in question. Let us describe the planning algorithm.

Let t_i^{Exit} is the predicted time of achievement of the exit point P_i of the current sector X_i, H_{ij} is the echelon used by aircraft in the point P_i and t_{i+1}^{Exit} is the predicted time of achievement of the exit point P_{i+1} of the next sector X_{i+1}.

The aircraft will transit into sector X_{i+1} at the time t_i^{Exit} if

A1. Free echelon will exist in the last leg of the sector X_{i+1} at the time t_{i+1}^{Exit} and this echelon $H_{i+1,k}$ is such that $H_{i+1,k} \leq H_{ij}$ and $H_{i+1,k}$ is an admissible echelon for movement in the point P_{i+1};

A2. Conflict–free plan of movement of the aircraft in question from the point $< P_i, H_{ij} >$ to the point $< P_{i+1}, H_{i+1,k} >$ is found.

Additionally, if several echelons meeting the condition A2 exist than the preference is paid to the higher ones. If the sector X_{i+1} comprises *N* legs then plan has to use *K≤N* transitions to the lower echelons. This requirement is caused by the necessity to use only admissible echelons in every transition points.
If at least one of the above given conditions, A1 and A2, is not held then the aircraft has to use the holding area of the current sector X_i.

Variant Pl-2. Next sector belongs to the arrival zone and Conf(Next sector)=2

This sub-algorithm is used in the case if hijacked aircraft does not conflict with the normal aircraft in question within the next sector X_{i+1} under condition that the former will not change its trajectory while preserving the same course and speed as were used for forecasting its trajectory. But, if hijacked aircraft has changed these attributes then conflict can occur.

Thus, in variant *Pl-2*, the normal aircraft plans to transit in the next sector X_{i+1} is the following conditions are held:

B1. Minimal distance between the aircraft in question and any other aircraft operating within the sector X_{i+1} during the same time interval meets separation standards. In this case the aircraft in question will be able to transits into any other echelon in any time moment;

B.2. The total number of aircraft operating within the sector X_{i+1} within the time interval $[t_i^{Exit}, t_{i+1}^{Exit}]$ is less than the total number of admissible echelons in the point P_{i+1}.

If at least one of the above given conditions, B1 and B2, is not held then the aircraft has to use the holding area of the current sector X_i.

Variant Pl-3. Next sector belongs to the arrival zone and Conf(Next sector) =3

Pl-3 variant corresponds to the case if transition of the normal aircraft to the sector X_{i+1} results in definite conflict with the hijacked aircraft. The aircraft is prohibited to use the next sector and it has to use the holding area of the current sector X_i.

Variant Pl-4. Next sector belongs to the approach zone and Conf(Next sector)=1.

If the permission to entry into approach (next) sector is received the normal aircraft decide to transit into it according to the plan received from air traffic operator. Otherwise it uses the holding zone of the current sector.

Variant Pl-5. Next sector belongs to the approach zone and Conf(Next sector)=2.

In this case, when normal aircraft is ready to entry into approach zone X_{i+1}, between this aircraft and hijacked one a conflict may happen if the latter changes its movement in appropriate way. It such case, the aircraft is prohibited to use the next sector and it has to use the holding area of the current sector X_i.

Variant Pl-6. Next sector belongs to the approach zone and Conf(Next sector)=3.

If *Conf(Approach sector)=3* and *Approach sector*= X_{i+1} then normal aircraft is prohibited to entry and it has to use the holding area of the current sector X_i.

4.5. Permission for entrance into approach sector

According to the organizational principle proposed in this research (see section 3.4) air traffic control within approach sector is the responsibility of air traffic operators. In this project, for simulation purpose, the simplified model of air traffic operators' activity is used. The idea of this model is to constraint the time intervals between to subsequent use of the same approach scheme by different aircraft. These constraints are determined by the set of functions denoted below as ***TRF***. These functions are determined as follows.

Let ***SH*** is the set of admissible movement schemes determined within approach sector. Each of them begins with entry point and ends in particular runway. ***TRF*** function is determined as matrix function which element on <*i,j*> position corresponds to the particular function $trf(sh_i, sh_j)$, where sh_i, $sh_j \in$ ***SH***, i.e. to the set of admissible approach schemes. Each function $trf(sh_i, sh_j)$ present time interval between entry times of a pair of normal aircraft using schemes sh_i, sh_j. Let us note that these functions are not commutative, i.e. $trf(sh_i, sh_j) \neq trf(sh_j, sh_i)$. In the latter function, the first argument corresponds to the approach scheme used earlier than the scheme indicated as the second argument. The values of these functions may be particular for each particular airport. In this research these values are selected on the experimental basis.

Use of these functions makes it easy to solve the existence of conflict between a pair of normal aircraft entering approach sector.

On the other side, these functions simulate safety policy of aircraft via regulation of the time intervals between any couple of aircraft when permission to entry into approach sector is issued. Of course, these functions determine only constraints to be taken into account by safety policy ordering the aircraft according to entry time. Safety policy for this case is formed as a set of priority rules depending on

various attributes of aircraft and their movement attributes. The following attributes may be used as arguments of the priority rules:

- P_{Entry} –the entry point into approach sector;
- *sh* –approach scheme used by aircraft;
- t_{E1} – estimated time of achievement, by aircraft, the point P_{Entry};
- t_{E2} –estimated time of the second achievement, by aircraft, the point P_{Entry} after one use of the last holding zone the arrival sector;
- Class of aircraft;
- Current deviation (delay or wise versa) from the aircraft scheduled time;
- Fuel reminder;
- Etc.

It is important to note that permission is issued on the basis of meeting safety policy in regard to all aircraft currently situated within approach sector.

As concern safety policy when hijacked aircraft appears near or within approach zone, this task requires additional research and simulation. In current version the behavior patterns of hijacked within approach zone are not considered so far.

4.6. Chapter concluding comments

Chapter 4 describes developed airspace deconfliction algorithm addressing the specific conditions of the involved aircraft whose pilots are assisted by autonomous software agents. These agents provide distributed autonomous decision making implementing cooperation through trade-off negotiation within their community. Solution of this task is assumed by Task 4 of the Project Work plan.

5. Design Project of Multi-agent Airspace Deconfliction System

The Project is developed based of airspace model (see Chapter 2), proposed organizational structure of the air traffic control within airport airspace (see Chapter 3) and using airspace deconfliction algorithm described in Chapter 4.

5.1. Meta-model of Multi-agent Airspace Deconfliction System

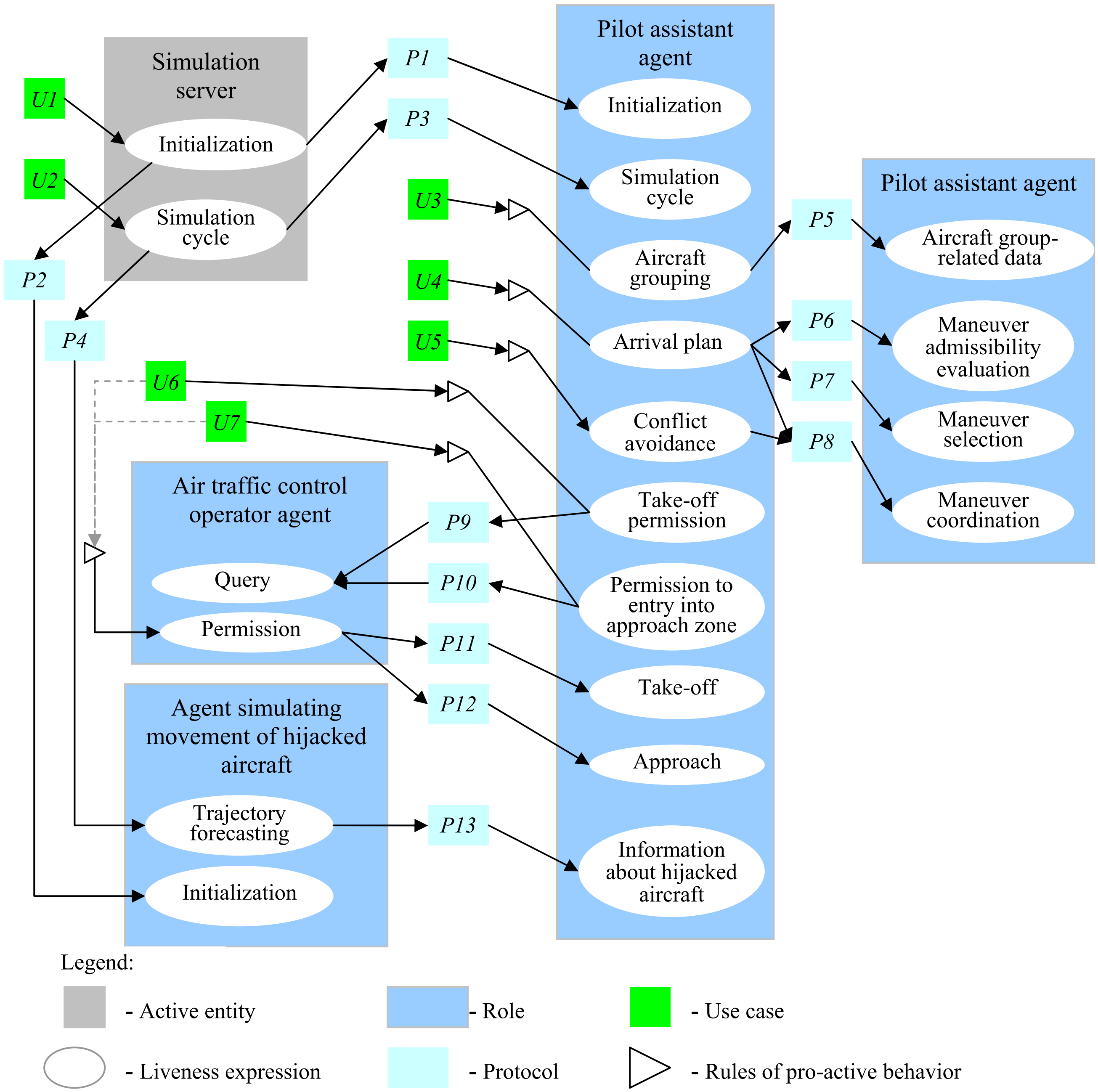


Fig. 5.1. Meta-model of Multi-agent Airspace Deconfliction System

Design project is specified based on some extension of Gaia methodology [Gaia], that is why below terminology used in Gaia is exploited. High-level conceptual model of the system in question is specifies in terms of diagram representing meta-model of the target system (Fig. 5.1). This diagram represents meta-model of the system in terms of roles to be allocated to agents and active software entities as well as

reflects interaction of the aforementioned components. In the Project each role is mapped an agent class performing it. Therefore, the terms "role" and "agent class" may substitute each other without misunderstanding. Hereinafter, the term "agent class" is mainly used.

In the developed project, three agent classes and single active software entity are introduced. Let us describe them.

Agent classes

- P*ilot assistant agent class* (*PA-agent class*)-agents of this class assist the pilots in deconfliction task solution;
- *Air traffic control operator agent class* (*ATCO-agent class*) – this agent class assists the air traffic control operator in decision making within approach sector;
- *Hijacked aircraft agent class* (*HA-agent class*) – this agent class intended for simulation and monitoring of hijacked aircraft movement.

Simulation server plays here the role of active program entity. It is intended for simulation of real time that is necessary to initiate real time events reflecting operating of entities involved in air traffic and air traffic control. Simulation server also provide interface to user, in particular, it supports the following functions:

- Visualization of the current air traffic situation within airport airspace
- Generation of the hijacked aircraft trajectory,
- Visualization of conflicts occurring between pairs of normal aircraft as well as between normal aircraft and hijacked ones.

According to Gaia methodology, formal specifications of agent classes (roles) are done in terms of so called *liveness expressions* . They specify the basic scenarios of agent classes' behavior in various tasks (use cases). In particular, specification of PA-agent class consists of 14 *liveness expressions* (*Initialization*, *Simulation cycle*, *Aircraft grouping*, *Arrival timetable monitoring*, ...) listed in Fig. 5.1. *ATCO agent class* model consists of two *liveness expressions* LE, i.e. *Query* and *Permission.* Agent class simulating movement of hijacked aircraft includes also two *liveness expressions* that are *Initializatio*n and *Trajectory forecasting*.

The subsequent description of the design project of the target system contains detailed description of every liveness expression. These descriptions are done in context of the Use cases in which corresponding liveness expression is involved. Let us remind that a "Use Case" is understood as one of target tasks to be solved by the multi-agent airspace deconfliction system. In the developed model, seven such use cases (tasks of the system as a whole), U1-U7 (see fig.5.1) are identified:

- (*U1*) Initialization of *PA-agent* class instances and initialization agent simulating movement of hijacked aircraft;
- (*U2*) Execution of simulation cycle
- (*U3*) Grouping of aircraft (instances of PA-agent class) that is used to decrease total information exchange traffic and to provide less computational complexity of the airspace deconfliction algorithm;
- (*U4*) Autonomous planning of own movements within arrival (zone) sectors by PA-agent class instances;
- (*U5*) Re-planning of own movements within arrival (zone) sectors by PA-agent class instances in order to avoid conflicts with hijacked aircraft.
- (*U6*) Normal aircraft' take–off control;
- (*U7*) Control the arriving aircraft during the time of their movement within approach zone (from the time when aircraft requests permission to entry approach zone till landing)

While performing the aforementioned tasks (use cases, scenarios) corresponding agent instances implement behavior specifies in terms of liveness expressions.

Two variants of liveness expression initiation are used:

- Agent initiates running of a liveness expression in response to an input message. For example (Fig. 5.1), PA- agent class instance initiates running of the liveness expression "*Take-off*" after receiving the input message from ATCO-agent class containing "*Take-off permission*".
- Agent itself initiates a liveness expression as a result of its pro-active emergent behavior, i.e. as a result of occurrence of some events within the environment. For example, when PA-agent class instance initiates running of the liveness expression "*Grouping*" after transition of the normal aircraft from a sector to another one.

5.2. Simulation Server

Input data of the server specifying air traffic situation within airport airspace are comprised the following data structures:

Specification of the normal aircraft approaching to the airport airspace but situated outside of it so far

- Entry point of the aircraft and echelon;
- Time of crossing the entry point;
- Aircraft class;
- Airport and target runway.

Specification of the normal aircraft departing the airport according to timetable

- Airport and take-off run way;
- Take-off time;
- Aircraft class;
- Exit point of aircraft from airport airspace.

Specification of hijacked aircraft consists of specification of two classes of events (a) Fact of appearance of hijacked aircraft within airport airspace and (b) Fact indicating modification of hijacked aircraft trajectory (speed, course). Both events are attributed as follows:

- Horizontal coordinates
- Altitude of modification of it;
- Course, Aircraft class.

The time of hijacked aircraft appearance is simulated randomly or introduced by user via corresponding user interface of the simulation server.

An important notice is that the values of speeds of normal aircraft as well as hijacked ones are determined in the simulation process using aircraft class and its speed interval depending on the altitude (see table 2.1 in section 2.2).

Every simulation cycle is run according to the algorithm which structure is depicted in Fig. 5.2.

Duration of the simulation cycle *dt* is constant and is determined using through interface. It can be changed at any time of simulation running.

Initialization of PA-agent class instances and hijacked aircraft agent instances is performed by server itself according to the records done in data base in advance. This distributed procedure is performed in accordance with *P1* and *P2* protocols.

Simulation cycle is done according to P3 and P4 protocols when the server sends to corresponding instances of PA-agent class and Hijacked aircraft agent class the message indicating duration of simulation cycle at the forthcoming simulation cycle.

Based on data received from the instances of PA-agent class and Hijacked aircraft agent class simulation server executes two tasks:

1) Checking meeting of the separation standards between pairs of normal aircraft as well as between normal aircraft and hijacked one(s);

2) Updates visualization of the current situation through user interface while reflecting the facts of separation standards violations if any at the previous simulation cycle.

5.3. Initialization of instances of PA-agent class and Hijacked aircraft agent class

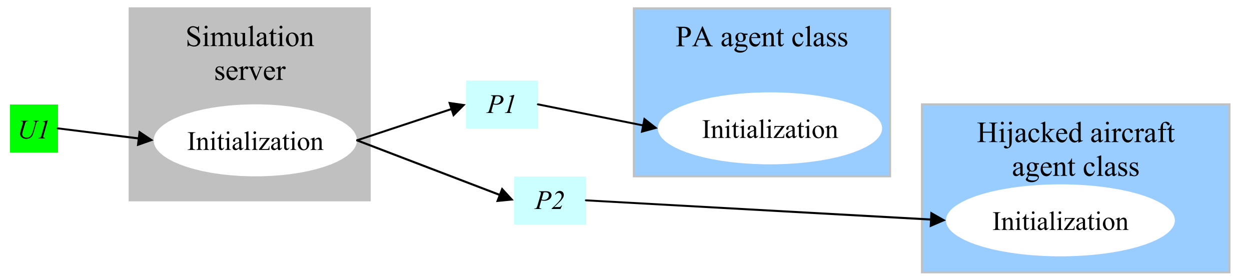


Fig. 5.3. Use case: Initialization of agent instances

5.3.1. PA-agent class: Liveness Expression *Initialization*

Events initializing running of liveness expression

Input message received according to *P1* protocol

Behavior scenario

1. Select the flight scheme of normal aircraft within airport arrival zone which is later used for planning and computing the exact temporal trajectory of aircraft movement. The flight scheme is computed using attributes of the aircraft and particular topology of the airport airspace in the area of aircraft entry point.

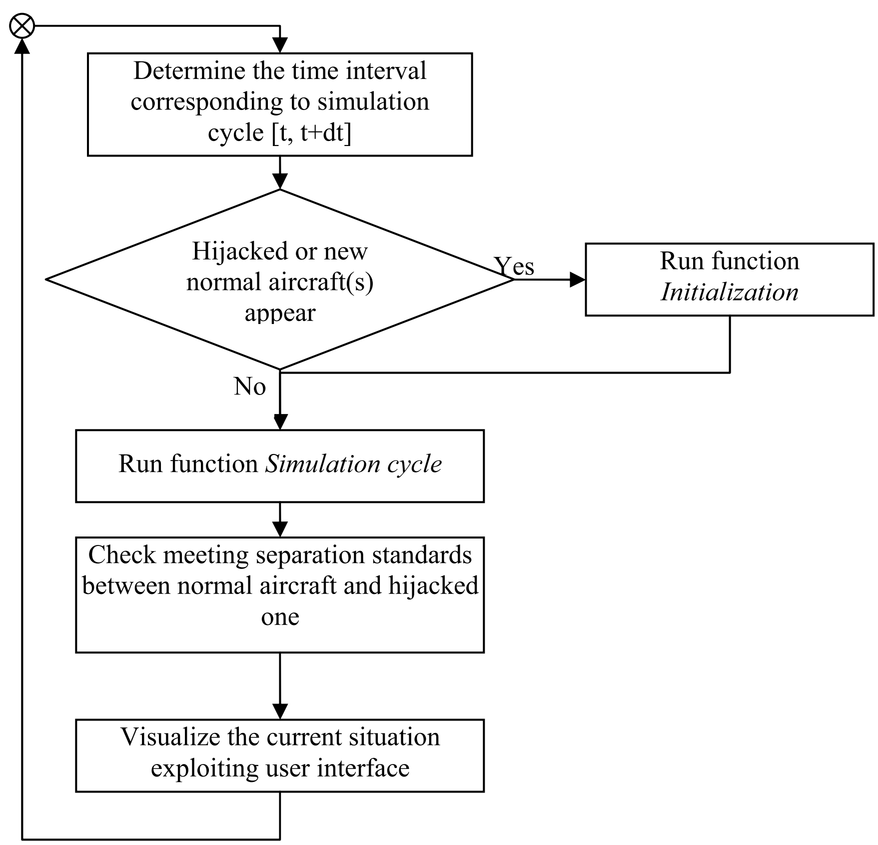


Fig.5.2.Simulation cycle algorithm

2. Assign status value of the aircraft that, depending on current state can take values from the following set:
   - ✓ *“Take-off readiness”*

- ✓ *“Waiting for take-off permission.”*
- ✓ *“Approaching to the airport airspace”*
- ✓ *“Movement within arrival zone”*
- ✓ *“Reaching to the approach zone”*
- ✓ *“Waiting for permission to entry into approach zone”*
- ✓ *“Movement within approach zone”*
- ✓ *“Movement to the exit point of the airport air space”*

In the considered situation, the status may be assigned only one of two following values::

- ➢ *“Take-off readiness”*
- ➢ *“Approaching to the airport airspace”*

3. Determine the first sector X_1 along which the entering aircraft will move and generate the event "*Get status of the first grou*p". This event, in turn, initiates running of the "*Grouping*" liveness expression.
4. Wait the event "Status of the next group is assigned" which is initiated as a result of running the "*Grouping*" liveness expression.
5. Send to the server in reply message assumed by protocol *P1* informing about completion of the initialization.

5.3.2. Hijacked Aircraft Agent Class: Liveness expression *Initialization*

Event initializing run of liveness expression

Getting input message according to *P1* protocol

Behavior Scenario

1. Record and save data specifying current trajectory of hijacked aircraft to use them later for forecasting of the future trajectory of it.
2. Send in reply message to the server while informing it, according to *P2* protocol, about completion of the initialization procedure.

5.4. Simulation cycle

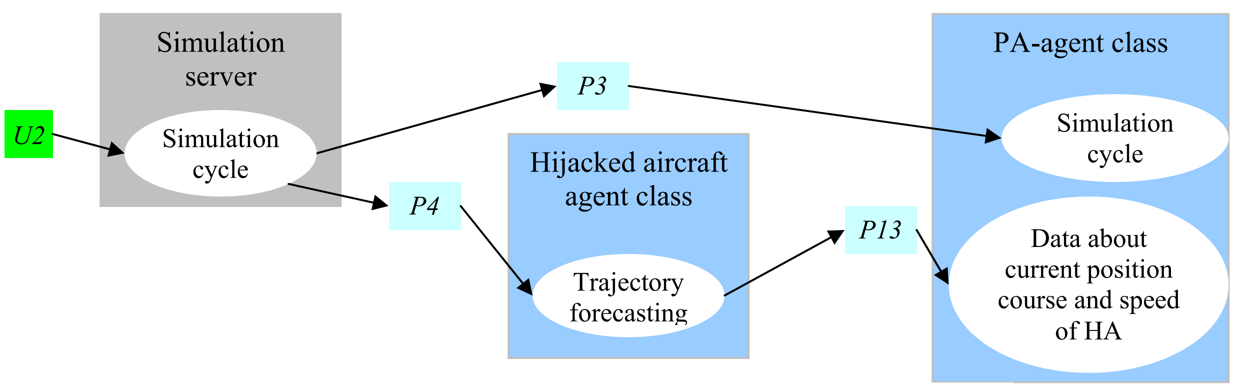


Fig. 5.4. Use case: Simulation cycle

5.4.1. PA-agent Class: Liveness Expression *Simulation cycle*

Event initializing run of liveness expression

Receiving an input message according to *P3* protocol

Behavior Scenario

Behavior scenario of liveness expression is comprised by two phases: (1) *Planning* and (2) *Movement simulation*. At the first phase, *Planning,* scenario is determined by the value of the Current State Status (CSS) of the normal aircraft.

Behavior scenario for the case CSS = *“readiness to Take-off”*

1. Generate event *“Request for take-off permission"*. Next, this event initiates run of liveness expression "Take-off permission".
2. Assign the CSS of normal aircraft the status value "*Waiting for take-off permission*".
3. Go to the phase "*Movement simulation"*.

Behavior scenario for the case CSS = *“Waiting for take-off permission ”*

1. Go to the phase "*Movement simulation"*.

Behavior scenario for the case CSS =*“Approaching to the airport airspace”*

1. Generate event *“To agree entry into airport airspace”*. This event initiate run of the liveness expression "*Arrival*".
2. Go in stand by state while waiting for one of the following two events: *“Entry into airport airspace is permitted"* or *“Entry into airport airspace is not permitted"*. Any of these events is generated as a result of running of the liveness expression *"Arrival plan"*.
3. If the message *“Entry into airport airspace is permitted"* is received then assign the CSS status value *“Movement within arrival zone"*.
4. Go to the phase "*Movement simulation"*.

Behavior scenario for the case CSS = *“Movement within arrival zone"*

1. Determine current and next movement sectors, X_i and X_{i+1} of the entered normal aircraft.
2. Compute the value of function *Conf( )* for the sectors X_i and X_{i+1}. If at least one of the functions *Conf(sector* X_i*)* or *Conf(sector* X_{i+1}) takes value "*3*" then generate event "*Conflict is possible*" that initiates running of the liveness expression "*Conflict avoidance*".
3. Go to the state of waiting for the event "*Plan is recomputed*" that should result from the run of the liveness expression "*Conflict avoidance*".
4. If safe plan for the sector X_i is not found then generate event "*Create movement plan*". This event initiates running of the liveness expression "*Arrival plan*".
5. Go to the state of waiting for the event *“Arrival plan is created”* resulting from completion of run of the liveness expression “*Arrival plan”*.
6. Go to the phase "*Movement simulation"*.

Behavior scenario for the case CSS =*“Reaching to the approach zone”*

1. Generate event *“Request for permission to entry into approach zone”*. This event initiates run of liveness expression *“Permission to entry into approach zone”*.
2. Assign CSS the value "*Waiting for permission to entry into approach zone”*.
3. Go to the phase "*Movement simulation"*

Behavior scenario for the case CSS =*“Waiting for permission to entry into approach zone”*

1. Go to the phase "*Movement simulation"*

Behavior scenario for the case CSS =*“Movement within approach zone”*

1. Go to the phase "*Movement simulation"*

Behavior scenario for the case CSS =*“Movement to the exit point of the airport air space”*

1. Go to the phase "*Movement simulation"*

Behavior scenario in the phase "*Movement simulation"*

1. If movement plan is created then run simulation within indicated simulation cycle; otherwise go to the item 5 below.
2. If in simulation procedure the exit point of the occupied sector is not found then go to the item 5 below.
3. Generate event "*Transition into next sector"*, that initiates running of the liveness expression *"Aircraft grouping"*.
4. Compute updated current and next movement sectors, X_i and X_{i+1}.
5. If the sector X_{i+1} belongs to the approach zone then assign CSS the value *"Reaching to the approach zone"*
6. If time before achievement of the exit point of the current sector is less threshold dT given as option, then generate event *"Time before exit from current sector is less dT"*. This initiates run of liveness expression *"Permission to entry into approach zone"*.
7. According to *P3* protocol send in reply message to the simulation server about completion of current simulation cycle.

5.4.2. Hijacked aircraft agent class: Liveness Expression *Movement Forecast*

Event initializing run of liveness expression

Arrival an input message according to *P4* protocol

Behavior Scenario

1. Run simulation using a) initial data received by agent at initialization procedure and b) designated time interval corresponding to the current simulation cycle.
2. Forecast the position of the hijacked aircraft for the time interval *dT* designated while assuming that it preserve course and speed value.
3. Compute the values of *Conf( )* function for all sectors according to algorithm described in section 4.3.
4. Using protocol *P13* inform all PA agent class a) forecasted trajectory of the hijacked aircraft, and b) values of *Conf( )* function for the sectors computed above in item 3.
5. Using protocol *P4* send in-reply message to the simulation server while informing it about completion of the simulation cycle.

5.4.3. PA agent class: Liveness expression *Information about hijacked aircraft*

Event initializing run of liveness expression

Input message incoming according to *P3* protocol.

Behavior Scenario

1. Update data concerning forecasting of the hijacked aircraft movement and value of *Conf( )* function for the sectors to which the normal aircraft belongs.

5.5. Grouping

5.5.1. PA agent class instance: Liveness expression *Grouping*

Event initializing run of liveness expression

- Event "*Get the status of the first group*". This event is generated as a result of completion liveness expression "*Initialization*".
- Event "T*ransition into the next sector*" This event is generated when liveness expression '*Simulation cycle*" is completed.

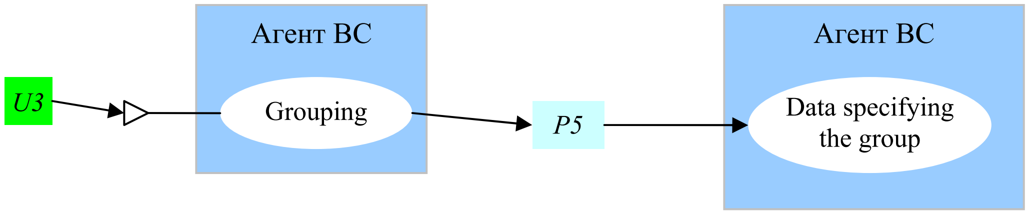


Fig.5.5. Use case: Grouping

Behavior Scenario

The behavior goal is to get the status of the group *G(Sector* X_i). The sector X_N is either the first sector of the airport airspace within which the entered normal aircraft will move or the next sector to which the aircraft will transit from the first one according to the assigned arrival scheme. The status of the group is assigned when (1) PA agent class instance "knows" about all group members *G(Sector* X_i) and (b) all group members "know" about the former one.

The behavior scenario may be divided into two phases (Fig .5.6). At the first phase, the agent gets group *G(Sector* X_i) status. This is done by aircraft according to following procedure:

1. Determine identifier of the next sector X_i.
2. Announce about itself as about the group *G(Sector* X_i) member. via publication, on Yellow pages, announcement about the service *Inform(*X_i*)* that is *"Providing the service concerning own plan of movement within the sector* X_i".
3. Discover all the *G(Sector* X_i) group members that is done via search for all PA agents class instances providing the same service.
4. Send to all the *G(Sector* X_i) group members information specifying its movement within the sector X_N. To that moment the single such attribute is computed by the aircraft that is the forecasted earliest time of its exit from the current sector which coincides with the time of its entry into the next sector (according to its plan that has to be computed before current time).
5. Go to the state of waiting of the event "*Status of the new sector is got*". This event is generated as a result of completion of the liveness expression "Aircraft group-related data" concerning *G(Sector* X_i) group.

After completion of the first phase, fulfillment of the second one starts. It is lasting till aircraft transit from the current sector X_i into the next one. During the above period, the PA agent class is continuing to fulfill its duties assumed for a group member while providing the service *Inform(*X_i*)* while the following events income:.

- *"New G(Sector* X_i*) group member"*. This event is generated by the liveness expression "*Aircraft group-related data*". In these cases, PA agent class instance sends information about own movement plan to new member only.
- *Update of the own movement plan.* This event is generated during fulfillment of the following liveness expressions: "*Arrival plan*", "*Conflict avoidance*", "*Approach*", "*Take-off*". In these cases, information is sent to all group members while informing the latter about updating of own plan.
- "*Transition into next sector*". This event is generated by liveness expression *"Simulation cycle"* .

5.5.2. PA agent class: Liveness expression *Aircraft group-related data*

Event initializing run of liveness expression

Receiving an input message using *P5* protocol

Behavior Scenario

Behavior Scenario is determined by the data sender and received data contents.

If data is received from new *G(Sector* X_i) group member then

1. Safe the data received.
2. Generate event "*New group member*". This event is processed within liveness expression "*Grouping*" which initiates sending, to new group member, information concerning corresponding aircraft current movement plan.

If data is received in reply of own registration within group then

1. Safe the data received.
2. If in-reply data received from all the group members then generate the status of the *G(Sector* X_i) group value and generate event "*Group status is assigned*". This event is next processed in the

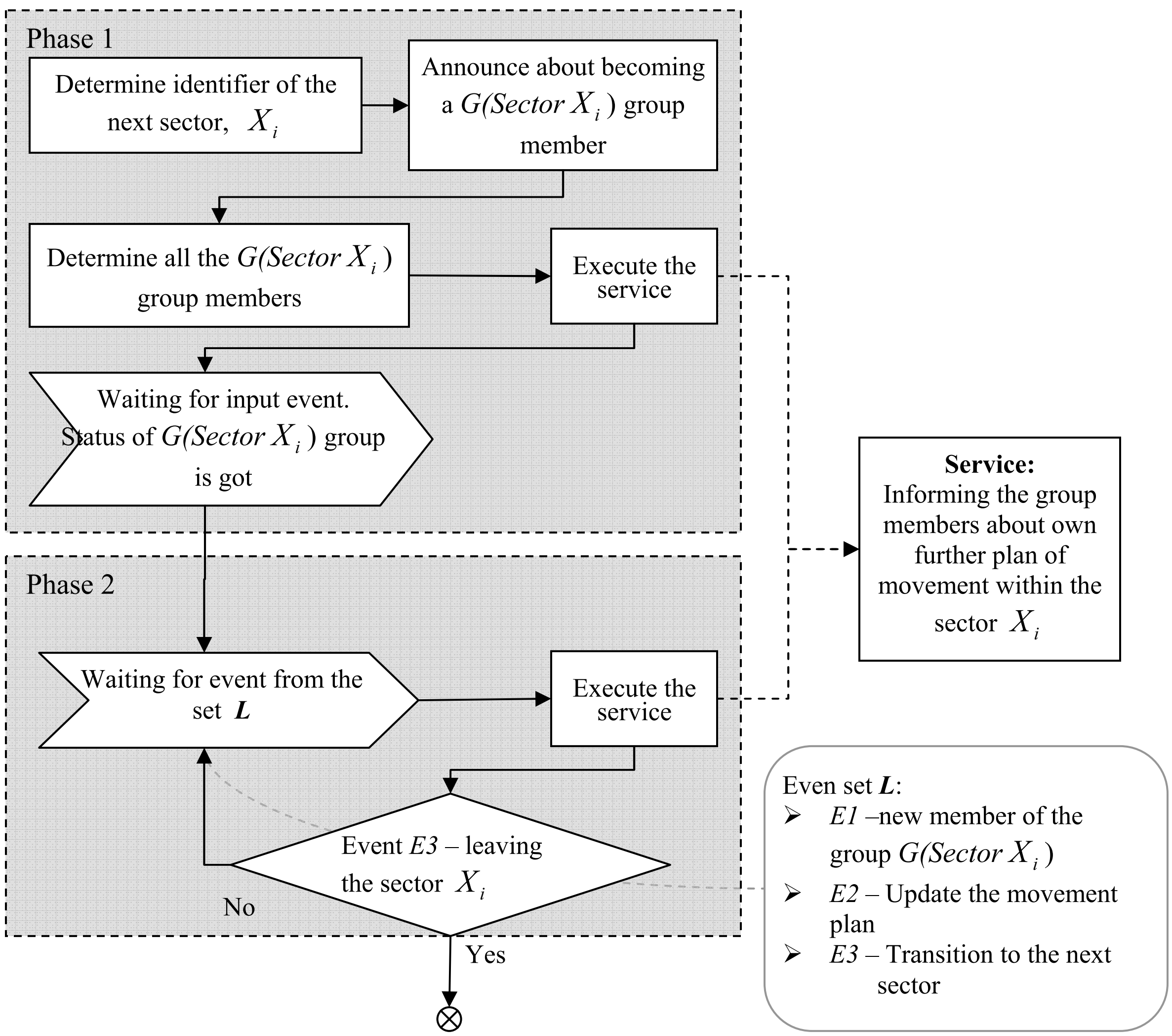


Fig. 5.6. Structure of the scenario of the liveness expression "*Grouping*"

liveness expression "*Grouping*".

If data is received from a *G(Sector* X_i) group member to inform about updated plan then

3. Safe the data received.

If data is received from a *G(Sector* X_i) group member to inform about its exit from the group then

1. Delete all the data records concerning it.

5.6. Arrival Plan

5.6.1. Preconditions

General description of the Use Case

This use case the normal aircraft plan their movement in the next sector. Two main tasks are solved in this use case:

- Coordination of agents behavior determining the planning order, and
- Planning algorithm itself.

The following rules coordinate the above mentioned agent behavior:

(1) Independent planning within different groups of sectors;

(2) Planning within a sector X_i is done in the following way:

a. The *G(Sector* X_i) sector group is split into three subsets that are:

- G_{IN} *(Sector* X_i) – the subset of the aircraft located inside the sector in question and having own movement plans, for this sector, developed.
- G_{HP} *(Sector* X_i) – the subset of aircraft intended to entry into the sector X_i, that do not so far entry into it but the corresponding plans are got ready.
- G_{PL} *(Sector* X_i) – the subset of aircraft that do not so far developed their plans of movement within the sector X_i.

The task concerning planning of aircraft movement within the sector X_i is the subject of efforts of the aircraft $Y_j \in G_{PL}$ *(Sector* X_i). Each such aircraft computes its plan only after ordering procedure determining, for the latter, when it is permitted. This order is built according to the set of rules described above in section 4.3. It starts plan computing when all the aircraft of the G_{PL} *(Sector* X_i) group having higher priority have already completed the planning and the aircraft in question is informed about their plans according to *P8* protocol.

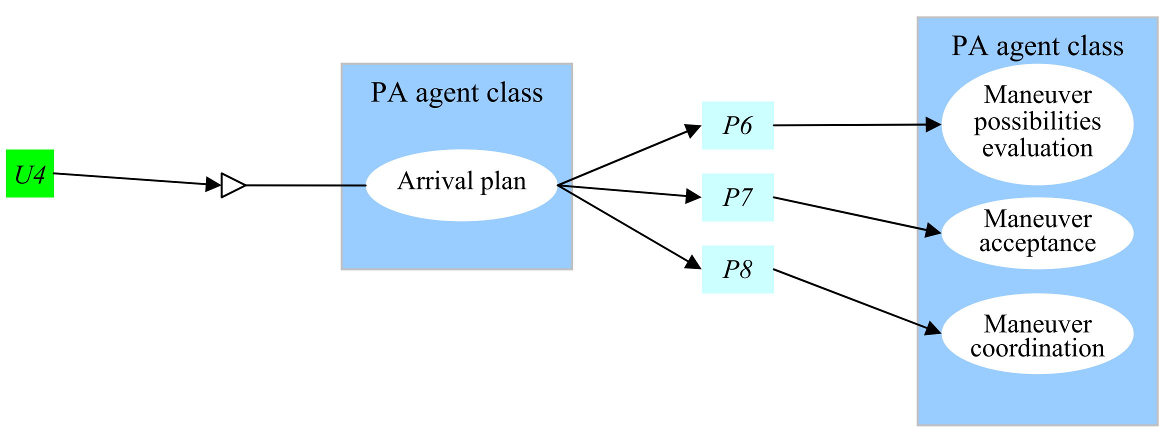


Fig. 5.7. Use case: "*Arrival Plan*"

Below, in description of the formal planning algorithm, the following denotations are used:

S_C –current sector o0f the aircraft location;

S_N –the next sector for which the planning is carried out;

$Plan(S_C)$, $Plan(S_N)$ – movement plans of the aircraft for the sectors S_C and S_N respectively;

$Leg(S) =< leg_1, leg_2,..., leg_r >$– sequence of legs, constituting the sector S ;

$Alt(S) = < n_1, n_2,..., n_r >$ – altitude-related "structure" of the sector S , i.e. the numbers of the echelons of the legs that are permitted to use in the end points of the corresponding legs.

$Plan(S) = < k_1, k_2,..., k_r >$ – scheme of the echelon changing within the sector S indicating, in descending perspective, what number of echelon is "lost" by aircraft within every leg of the sector. For example, is a sector comprises three legs then the sequence < *0, 1, 2* > indicated that, in the first leg, the aircraft is moving horizontally, in the second one it descends to the next lower echelon and in the third leg in has to descend in two echelons.

In order to have a possibility to compare a pair of descending – related scheme of an aircraft descending movement subject that the total number of the "lost" echelons is preserved the same a preference function is used below. It is said that a descending – related scheme is better than the second one if more echelons are "lost" in the later legs. For example, descending – related scheme <*0, 1, 2*> is more preferable ("better") than the scheme < *1, 1, 1*>.

$Exit(S)$ –the total number of echelons that are permitted for use in the end point of sector S .

Let us consider additional safety conditions to be met in planning of the aircraft movement within a sector S_N if the aforementioned aircraft has *Conf( )* function value equal to *2*.

Condition 1. If difference of the altitudes used by normal and hijacked aircraft is more than 600 meters then this pair is conflict-free independently on the sector occupied by the latter. Formally this condition looks as follows:

> *If the safety standards determined in horizontal projections of two aircraft, normal and hijacked ones, are not met then, to avoid conflict, two next higher and two next lower echelons in regard to the hijacked aircraft are prohibited for use by corresponding "normal" aircraft.*

Accordingly the following condition has to be met for the aircraft of the sets $G_{IN}(S)$ and $G_{HP}(S)$:

$$|G_{IN}(S)|+|G_{HP}(S)|< Exir(S) -4.$$

If the above condition is met the normal aircraft of the sector S potentially can choose free echelon in holding area if the hijacked aircraft is moving near this zone or even intersects it.

Condition 2. Minimal distance between any two aircraft of a sector, in horizontal projection, cannot be less than 20 km independently of the echelons they occupy.

This condition provides conflict free movement for any normal aircraft and at any time if it transits to any admissible echelon

5.6.2. PA agent of normal aircraft: Liveness expression *Arrival Plan*

Events initiating liveness expression running

- Event "*Agree the entrance into arrival zone*". This event is generated by the liveness expression "*Simulation cycle*".
- Event "*Compute arrival plan*" It is generated by the liveness expression "*Simulation cycle*" as well.

Behavior scenario

The scenario specifies planning procedure

1. Compute the own aircraft priority among other aircraft of $G_{PL}(S)$ group. The needed input data are received by the group aircraft via information exchange procedure.
2. If an aircraft has not the highest priority, wait for the event *"The right for decision making is granted"*. This event is initiated by liveness expression "*Maneuver coordination*".
3. If variable "*Transition into next sector*" has value "*Mandatory*" then go to step 6.
4. If, for sector S_N, *Conf( )=3* then add to $Plan(S_C)$ the command to use sector holding zone and stop the planning.
5. If, for sector S_N, *Conf( )=2* then check the status of the Condition 1. If transition to the sector S_N violates this condition then add to $Plan(S_C)$ the command to use sector holding zone and stop the planning.
6. Compute the set of altitude-related schemes *Alt_Scheme*={< $S\_Plan(S_N)$, Y_j> for aircraft Y_j, $j \in G_{PL}(S_N)$ while restricting the descending speed to fixed value, for example, to 5 meters/per second.
7. Select the most preferable descending–related scheme for aircraft. Let us denote the selected scheme $S\_Plan(S_N^{selected})$.
8. Based on selected descending–related scheme, $S\_Plan(S_N^{selected})$, the aircraft in question develop the plan. Selection of planning procedure depends on the value of *Conf( )* function *for the aircraft within sector* S_N:

   If, for sector S_N, *Conf( )=1* then use procedure *A_Scheduling;*

   If, for sector S_N, *Conf( )=2* then use procedure *B_Scheduling;*

As a result, either a plan $Plan(S_N)$ is developed, or an admissible plan does not exist.

It is important to note that if $Plan(S_N)$ computed via use of *B_Scheduling* procedure exists then it is also exist in case of use of *A_Scheduling* procedure. The inverse is not the case.

9. If the plan $Plan(S_N)$ is found then go to item 13.
10. If the plan $Plan(S_N)$ is not found then delete plan scheme $S\_Plan(S_N^{selected})$ from *Alt_Scheme*.
11. If the resulting set *Alt_Scheme* is not empty then go to item 7. Otherwise go to item 12.
12. If aircraft Y_j is already located in arrival zone then add into the plan the command to use the sector's holding zone; otherwise, increase the planned the aircraft planned entry time into arrival zone at simulation cycle duration while increasing the Delay attribute through adding to its value the duration of the simulation cycle.
13. Generate event "*Change of own movement plan*". This event initiates "*Grouping*" liveness expression.
14. Find the PA agent having currently the highest priority. If such agent exists then send to it the permission to start planning procedure using protocol *F8*.

A_Scheduling Procedure

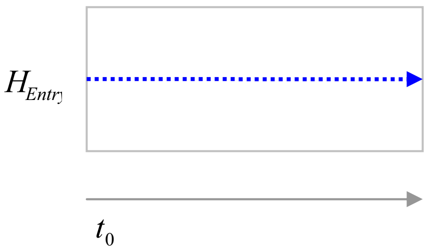
1. Create *AP_sector_List* containing the PA-agent instances which have to agree their plans (*Off-path jogging* maneuvers on particular legs).
2. If planning procedure succeeded for all legs of the S_N sector then go to item 9; otherwise, select the subsequent leg of the scheme S_N. For the latter, the initial data are as follows
   - H_{Entry} – an echelon corresponding to the entry point of the leg used by aircraft;
   - H_{Exit} an echelon corresponding to the exit point of the leg used by aircraft according to the altitude–related scheme;
   - t_0 – leg entry time. For the first leg of the sector S_N this time corresponds to the assumed arrival zone entry time. For the subsequent legs this times are formed within planning procedure.
3. Find subset of $G_{PL}(S_N)$ group that may conflict with each other while moving within the leg of the sector S_N. Let us denote this subset of aircraft as *AP_Set* $\in G\ (S_N)$. The latter subset is composed of the aircraft $Y_j \in G\ (S_N)$ that
   - Approaching to each other at a too small horizontal distance, i.e. that is less than it is permitted by separation standards;
   - While moving within the leg, belong to the same echelon H_{Entry} and/or H_{Exit} at some time intervals or cross them in transition maneuver.

   To find the subset *AP_Set* the PA agents using the initial data received via mutual information exchange.
4. Each aircraft of the subset *AP_Set* try to find conflict free plan of movement within the leg in question using the information received via mutual information exchange as well. This task is explained in Fig, 5.8 (a). According to the selected scheme $S\_Plan(S_N)$ for leg in question, two variants of movements can take place, i.e. movement in the same echelon and descending movement.
5. Complete the procedure while returning result "*For* $S\_Plan(S_N)$ *no admissible solution exists*" if
   - ✓ Either for one of the conflicting aircraft from subset *AP_Set* no decision exists,
   - ✓ The time intervals $[t_{entry}, t_{exit}]$ for the conflicting pair of the aircraft determining admissible time of transition in other echelon are not overlapping.
6. Define the list of aircraft, *AP_leg_List* that have to be taken into account in agreement of *Off-path jogging* maneuver.
7. If the list *AP_leg_List* is empty then go to the item 1.
8. If the list *AP_leg_List* is not empty then send to PA-agent instances of aircraft contained in this list the requirements to be met during fulfillment of *Off-path jogging* maneuver. Sending is done according to *P6* protocol. The aforementioned requirements formulate the time intervals when corresponding aircraft has to fulfill its vertical maneuver while preserving 5 km distance from the leg axis in order to meet the separation standards.
9. The sender can receive two types of in-reply messages from the aircraft forming the list *AP_leg_List*, that are (1) maneuver is possible or (2) maneuver is impossible. If at least one of the aircraft reply that maneuver is impossible then the procedure is ended with return "*For altitude-related scheme* $S\_Plan(S_N)$ *there is no admissible solution*". Otherwise add the list *AP_leg_List* to the *AP_Set* list and go to item 1.

10. If the list *AP_sector_List* is not empty then, using *P6* protocol, send messages to the agents of this list while informing them about adding previously agreed solutions concerning *Off-path jogging* maneuvers in their movement plans.

11. Complete the procedure while returning the result, i.e. *computed* $Plan(S_N)$.

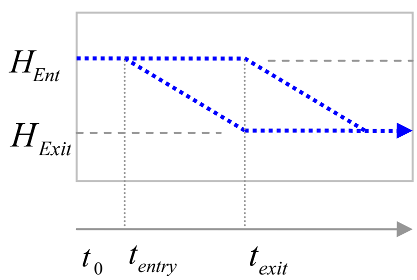
Movement within the leg using the single echelon



Possible cases:

- There is no conflict;
- Conflict will not happen if both aircraft agree the *Off-path jogging* maneuver
- Conflict–free plan is not exists.

Movement within the leg using echelon change



Possible cases:

- There is no conflict;
- There is no conflict if)
  - ✓ Transition can be started within time interval $[t_{entry}, t_{exit}]$
- Both aircraft can agree the *Off-path jogging* maneuver
- Conflict–free plan is not exists.

Fig. 5.8 (a).Development of the movement plan which is conflict free with regard to other aircraft

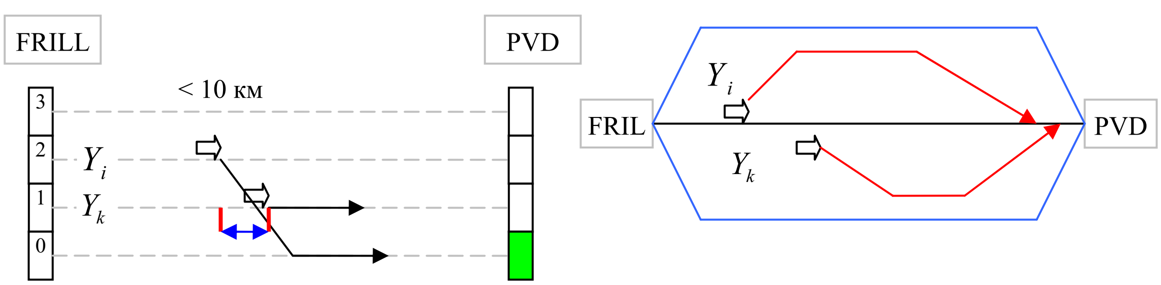


Fig.5.8 (b). Graphical explanation of the *Off-path jogging* maneuver

B_Scheduling procedure

1. Check meeting of *Condition B*.

2. If it is met then create movement plan $Plan(S_N)$. Plan creation consists in detection of the time instants corresponding to start of transitions of aircraft at corresponding echelons assumed by echelon-associated schemes of the aircraft. An important is to find the latest admissible time of the transition begin and select it as final decision.

5.6.3. PA-agent class: Liveness expression *Maneuver admissibility evaluation*

Events initiating liveness expression running

Receiving an input message according to *P6* protocol.

Behavior scenario

Input message contains information about time interval $[t_1, t_2]$ within which the aircraft has to carry out the maneuver in vertical plane that is 5 km away from the axis line of the corresponding leg that is necessary to meet the separation standard. While using this data, PA agent class instance create the plan of *Off-path jogging* maneuver. The task is explained in Fig. 5.9 a) and 5.9 b). The first figure corresponds to the case when aircraft is initially moving horizontally within the leg. The second figure explains the case when the aircraft has to execute *Off-path jogging* while moving with descending to occupy the lower echelon of the leg. In both cases the task is to compute the time instants t_{start} and t_{end} determining time of begin and end of the maneuver.

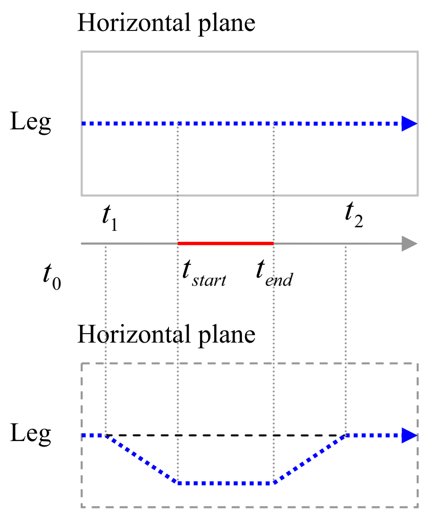


Fig. 5.9 a). *Off-path jogging* maneuver

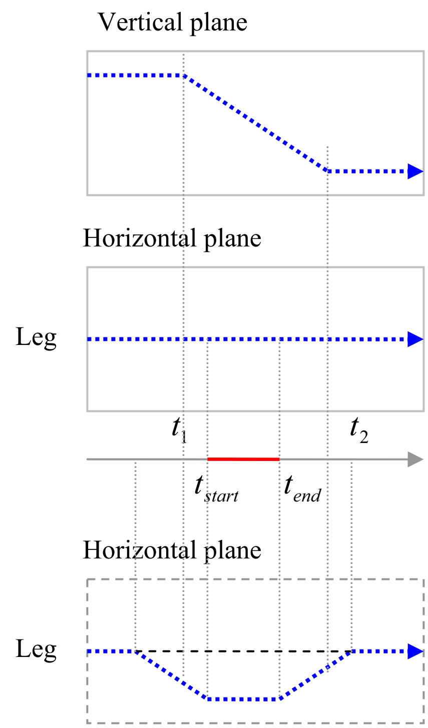


Fig .5.9 b). *Off-path jogging* maneuver

PA agent class instance refuses the proposed maneuver in the following two cases:

- If it already contains a *Off-path jogging* maneuver agreed previously with other aircraft (since the latter has higher priority);
- If the aircraft at the time instant t_{end} is located "too close" to the leg exit point that can lead to the impossibility to return to the leg axis at its exit point that is mandatory requirement of the airport airspace usage regulation

Accordingly, the behavior scenario is as follows:

1. Detect whether planned maneuver is admissible.
2. If it is admissible then compute its time-related attributes and save the result without changing the plan itself in order to further agree it with other aircraft. The final solution is made by the PA agent class instance that initiates the *P7* protocol. Its decision has to be sent according to *P8* protocol.
3. The decision made by the aircraft (admissibility or not admissibility of the maneuver) it send using P7 protocol.

5.6.4. PA-agent class: Liveness expression *Maneuver acceptance*

Events initiating liveness expression running

Receiving input message according to *P7* protocol.

Behavior scenario

PA-agent class instance includes the command to carry out the agreed *Off-path jogging* maneuver into own movement plan.

5.6.5. PA-agent class: Liveness Expression *Maneuver coordination*

Events initiating liveness expression running

Receiving input message according to *P8* protocol.

Behavior scenario

Generate event "Received the right to make decision". This event initiates continuation of the running of the liveness expression "*Arrival plan*".

5.7. Conflict Avoidance

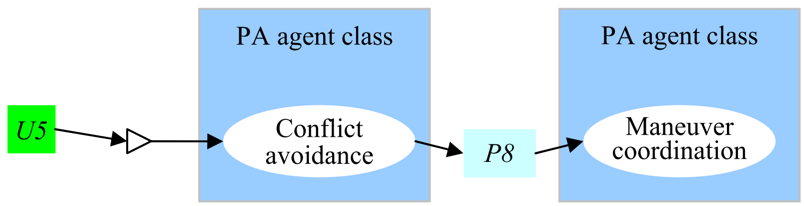


Fig. 5.10. Use case: Conflict avoidance

5.7.1. Conceptual description of the Use case

The objective of this Use case is correcting the plans of aircraft movement in the situations when a threat of conflict with hijacked aircraft happens. The rules according to which the aircraft behave in such situations are as follows:

Rule 1. The aircraft are prohibited to entry into the sectors having the value *3* of the function *Conf( )*;

Rule 1. If the target sector have the value *2* of the function *Conf( )* then transition into the sector is permitted only if *Conditions 1* and *2* (see subsection 5.5.1) are met.

The threat of conflict with hijacked aircraft may happen only in the case when either normal aircraft is moving within the sector with function *Conf( )*=3 or its plan assume such variant.

If rules 1 and 2 are met then the sector function *Conf( )=3* may be possible in the case if hijacked aircraft changes its course in the ways when the latter starts to approach to the sector within which the normal aircraft is either already located or approaching the next sector of the aircraft plan.

Actually these rules prevent conflict occurrence. In particular, before getting the value *3* the *Conf( )* the sector is assigned the value equal to 2, and that is why the *rules 1* and *2* if *conditions 1* and *2* are met form a *sparse* space and this makes it possible to remarkably decrease the negotiations (or avoid it at all) intended to the needed plan changes resulting in conflict avoidance. Actually, any aircraft located within a sector assigned *Conf( )*=3, due to "sparse space" is capable to find conflict free echelon of such sector. This idea is the basic one for the aircraft behavior corresponding to the liveness expression *Conflict avoidance*.

A possible case is when several aircraft find out within the sector having *Conf( )=3*. In this case, the plan coordination strategy is the same as in normal situations if the hijacked aircraft trajectory does not imply a threat in regard to several other sectors that may be used by aircraft. The coordination is carried out in accordance with the P8 protocol and liveness expression "*Maneuver coordination*" described in section 5.5.

5.7.2. PA agent class: Liveness expression *Conflict avoidance*

Events initiating liveness expression running

Event "Conflict is possible". This event is generated as a result of running of the "Simulation cycle"

Behavior scenario

Let S_C and S_N be the current and the next sectors of aircraft movement respectively. A variant of the plan change is selected depending on the existing situation class and conflict type. It is considered the following three classes of the situations:

1. Case_23: $Conf(S_C) = 2$ & $Conf(S_N) = 3$;
2. Case_33:. $Conf(S_C) = 3$ & $Conf(S_N) = 3$;
3. Case_32: $Conf(S_C) = 3$ & $Conf(S_N) = 2$.

And four possible variants of the conflicts depicted in Fig. 5.11.

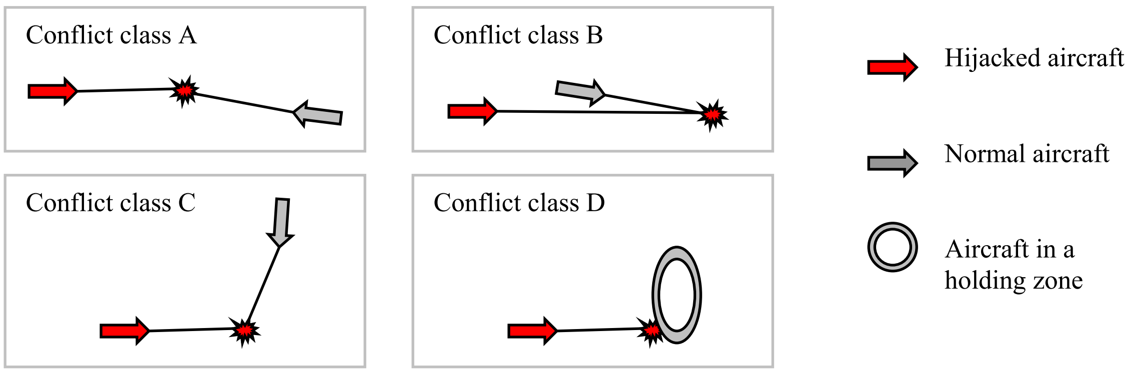


Fig.5.11. Variants of conflicts between normal and hijacked aircraft

Using such information, PA agent class instances refine their intentions to transit from the sector S_C into the sector S_N using the following rules:

- Rule 1. If the situation corresponds to the Case_23 PA agent class instance refuses from its intention to transit into the sector S_N and is waiting when the situation changes and makes decision to transit into the holding zone.
- Rule 2. If the situation corresponds to the Case_32 PA agent class instance has to urgently transit into the sector S_N and has not to decide to use the holding zone of the sector S_C.
- Rule 3. If the situation corresponds to the Case_32 PA agent class instance refines its intention depending on the conflict class. It preserves its intention to transit into the sector S_N if conflict of type A, C or D happens. If conflict of variant B happens then PA agent class instance refused from intention to transit into the next sector and has to decide to use the holding zone of the sector S_C.

Thus, the behavior scenario is as follows:

1. Determine priority of own aircraft among all the other ones currently located within the sector S_C. This is done via use of the data received via information exchange
2. If an aircraft is not of highest priority then transit into the state of waiting of the event "*The right to make decision is received*" and continue planning process after having this event received.
3. If the event "*The right to make decision is received*" incomes then continue the planning procedure. This event is generated as a result of running the liveness expression "*Maneuver coordination*".
4. Determine the leg (or legs if conflict of variant B happens) on which the conflict with the hijacked aircraft is forecasted.

5. If *Conf*(S_C) = 3 then cancel current movement plan *Plan*(S_C).

6. If *Conf*(S_N) = 3 then cancel current movement plan *Plan*(S_N).

7. Create the set $\{ S\_Plan(S_C) \}$ containing all potentially admissible schemes of the movement plans $S\_Plan(S_C)$ within the current sector S_C using the following three conditions:

   - For the legs of conflicts with the hijacked aircraft the selected echelons has to distant from the hijacked aircraft echelon at no less then 600 m;
   - For the last sector S_C leg does not use the echelons that have already been selected by the aircraft of higher priorities;
   - Vertical speed component in ascending / descending maneuver has to be equal to or less then the selected threshold, e.g., 5 m/per sec.

   It is worth to note that the set $\{ S\_Plan(S_C) \}$ already is not empty due to the developed rules.

8. Arbitrary select $S\_Plan(S_C)$ scheme and compute new movement plan *Plan*(S_C) within the current sector substituting the former one.
9. Using the rules *Rule 1*, *Rule 2* and *Rule 3*, assign the resulting value to the variable "*Transition to the next sector*" which can take one of two values, either "*Prohibited*" or "*Admitted*". This value is used for selection of the behavior scenario during running of the liveness expression "*Arrival plan*".
10. Generate event "Movement plan is corrected". This event initiates run continuation of the liveness expression "Simulation cycle".
11. Determine the normal aircraft situated within the sector S_C with the next highest priority and pass to the corresponding PA agent class instance the right to make decision using *P8* interaction protocol.

5.8. Entry into approach zone

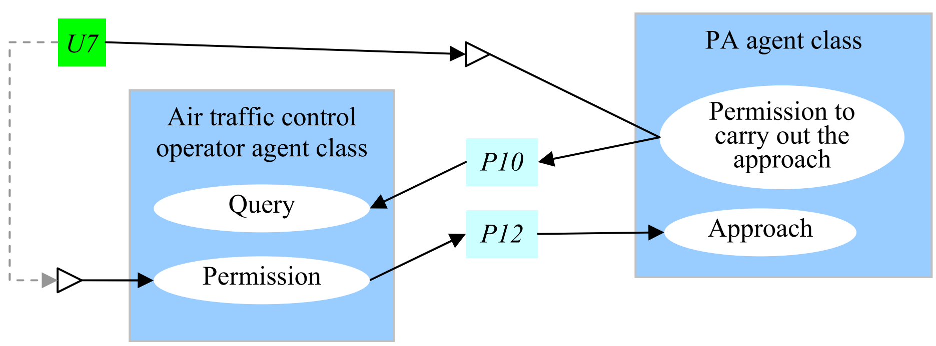


Fig.5.12. Use case: *Entry into approach zone*

5.8.1. PA Agent Class: Liveness Expression *Permission to entry into approach zone*

Events initiating liveness expression running

- Event "*Query the permission to entry into approach zone*". It is generated as a result of running liveness expression "*Simulation cycle*".
- Event "*The time interval remained to the current sector exit time is less than dT"*. It is important in the case when permission for use of approach sector in not received so far.

Behavior scenario

1. If the time interval remained to the current sector exit time is less than dT then add to the current plan the command to use the current sector holding zone.
2. Compute the earliest forecasted time when the entry into approach zone can be started.
3. Send to Air Traffic Control Operator (ATCO) agent class instance (repeated) query for permission to entry into approach zone using *P10* protocol. This query contains the following data:
   - P_{Entry}^{Appr} –approach zone entry point;
   - *Sh* – movement scheme within approach zone;
   - $t_{Entry-1}^{Appr}$ – computed time of achievement by aircraft the approach zone entry point P_{Entry}^{Appr} ;
   - $t_{Entry-2}^{Appr}$– computed time of the repeated achievement by aircraft the approach zone entry point P_{Entry}^{Appr} after use of the holding zone;
   - Aircraft class;
   - Current variation (delay or advance) from the timetable;
   - Fuel remained;
   - Etc.

5.8.2. Air traffic control operator agent: Liveness expression *Query*

Events initiating liveness expression running

Receiving the input event according to *P10* protocol.

Behavior scenario

1. Register the received query in the total list of the queries requesting permission to entry into the approach zone.
2. If repeated query is received then delete the previous one from the list.

5.8.3. Air traffic control operator agent: Liveness expression *Permission*

Events initiating liveness expression running

Event "*Time mark*" generated on regular basis in given time interval optionally determined by the user.

Behavior scenario

(Remark: This scenario is the same as described in section 4.5. It is briefly repeated)

1. For every query recorded in the total query list compute the earliest start time of the conflict–free use of the requested approach scheme. To solve this task, the following data are used: (a) Data about aircraft that are already situated within approach zone; (b) RTF functions described in section 4.5.
2. According to the rules determining the order of the approach zone utilization, select the query of highest priority.
3. Send permission to the selected aircraft using *P12* protocol;
4. After receiving the in-reply message (according to *P12* protocol) delete the query from the total query list and update the data concerning aircraft operating within the approach zone while including the data concerning the newly entered aircraft.

5.8.4. PA agent class: Liveness expression *Approach*

Events initiating liveness expression running

Receiving the input message according to *P10* protocol.

Behavior scenario

1. Based on selected movement scheme compute the movement plan within approach zone.

2. Send in-reply message informing about receiving of the permission to entry into approach zone. This is done according to P10 protocol.

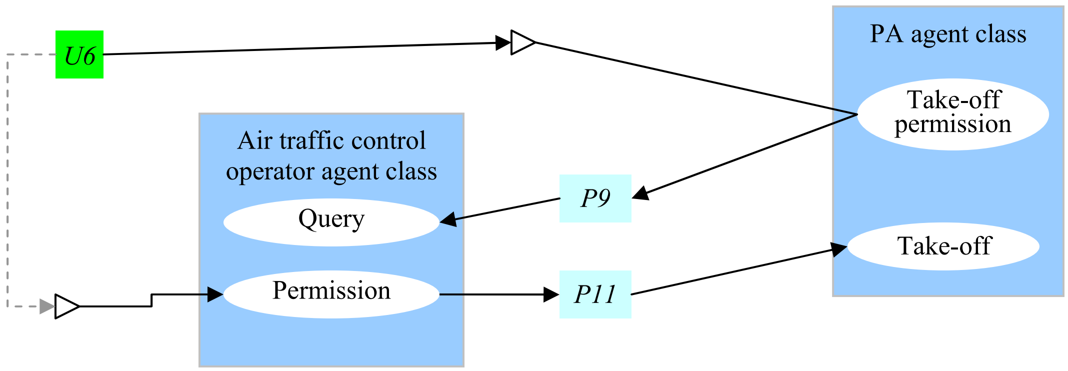


Рис.5.13. Use case: Взлет ВС

3. Assign the status variable SSC the value "*Movement inside the approach zone*".

5.9. Take-off

5.9.1. PA agent class: Liveness expression *Take-off Permission*

Events initiating liveness expression running

Event "*Query for take–off permission*" generated during run of the liveness expression "*Simulation cycle*".

Behavior scenario

1. Using *P9* protocol, send query for take–off permission. The message contains the following data:
   - $P_{rw}^{take-off}$ – take–off runway;
   - *Sh* – movement scheme within approach zone,
   - $T_{Take-off}$ – planned time instant of the take–off,
   - Current delay of the tale–off time.

   Description of ATCO behavior scenario represented by the liveness expressions "*Query*" and "*Permission*" is done in section 5.7.

5.9.2. PA agent class: Liveness expression *Take–off*

Events initiating liveness expression running

Receiving the input message according to *P11* protocol.

Behavior scenario

1. Using assigned scheme of movement in approach zone, create concrete movement plan.
2. Using protocol *P11send* in reply message about receiving permission for take–off.
3. Assign the status variable SSC the value "*Movement out of airport airspace*".

5.10. Chapter concluding comments

Chapter 5 carefully describes the developed design project of multi-agent airspace deconfliction system, specification of its basic components and their interaction. It includes specification of the multi-agent airspace deconfliction system meta–model and protocols representing architecture of the system in question (assumed by Task 6), model of the simulation server that has been used for verification and validation of the software implementation of the developed deconfliction algorithm (Task 5) and particular multi-agent airspace deconfliction system components (Task 8).

6. Graphical User Interface

6.1. Main Window

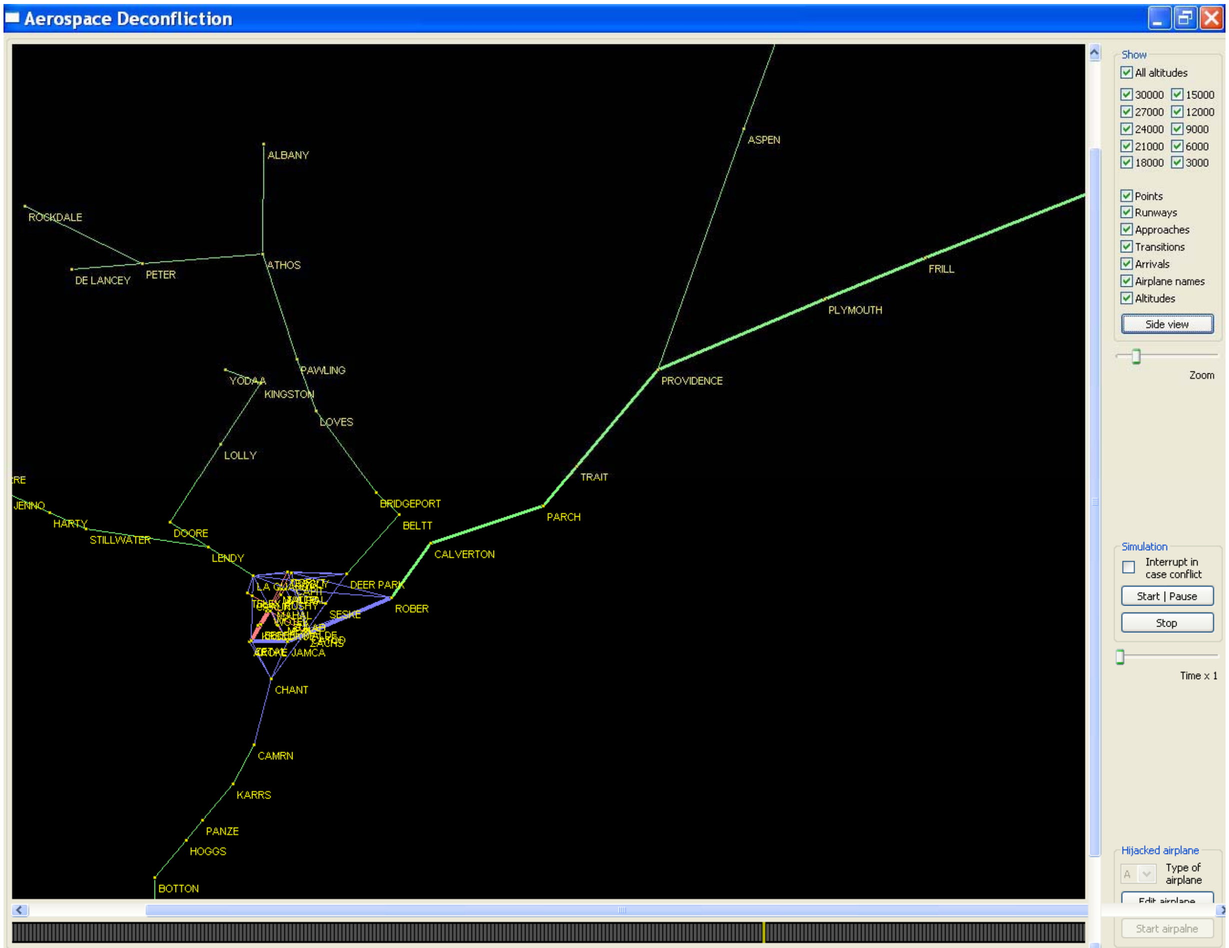


Fig. 6.1. Main window

Main window (Fig. 6.1.) is used for visualization of the followings:

- Airport airspace topology (horizontal projection);
- Current positions of the aircraft at simulation time instant, and
- Detected conflicts.

If at current time instant a conflict between a pair of the aircraft is detected then this conflict is depicted by red line connecting the conflicting aircraft. This is done only during time interval when the conflict exists. If it is eliminated the red line is deleted.

Interface also depicts some "statistics" of the detected conflicts. For this purpose, the sequence of the executed simulation cycles is depicted in the low part of the window, at that the cycles when conflict(s) exists are depicted in red color whereas conflict–free cycles are depicted in green color.

Actually program component supporting graphical interface operation performs safety norm checking and it does this independently of the agents' behavior. That is why this component may also be indirectly used for agent behavior validation.

Graphical Interface Options

- ✓ Image scaling;
- ✓ Optional filtering of data visualized on image;
- ✓ Altitude-based filtering of data represented as horizontal projection;

Simulation control: Functional capabilities

- ✓ Scaling of the simulation speed;
- ✓ Simulation process interruption in case if conflict happens;
- ✓ Simulation mode control: selection of continuous mode of simulation or cycle-based one. In the last mode, every simulation cycle is user–driven;
- ✓ Detection of time instant when hijacked aircraft appears.

Other control functions

- ➢ Selection of a movement scheme and visualization of aircraft movements in vertical plane (see section 6.2).
- ➢ Manual input of hijacked aircraft movement data (see section 6.3).

6.2. Visualization of Selected Movement Scheme in Vertical Projection

An example of visualization of a movement scheme–related situation in vertical plane is given in Fig. 6.2. In this figure, the arrival scheme corresponding to sequential movement through points *HANK, FRILL, PLYMOUTH, PROVIDENCE, TRAIT, PARCH, CALVERTON, ROBER* (see Fig. 6.1) of the JFK airport is depicted. In this picture, the aircraft situated in the "proximity" of the legs (at distances less than 5 km) constituting this scheme are depicted.

Horizontal lines represented in this window on background depict the echelons. In the window, only

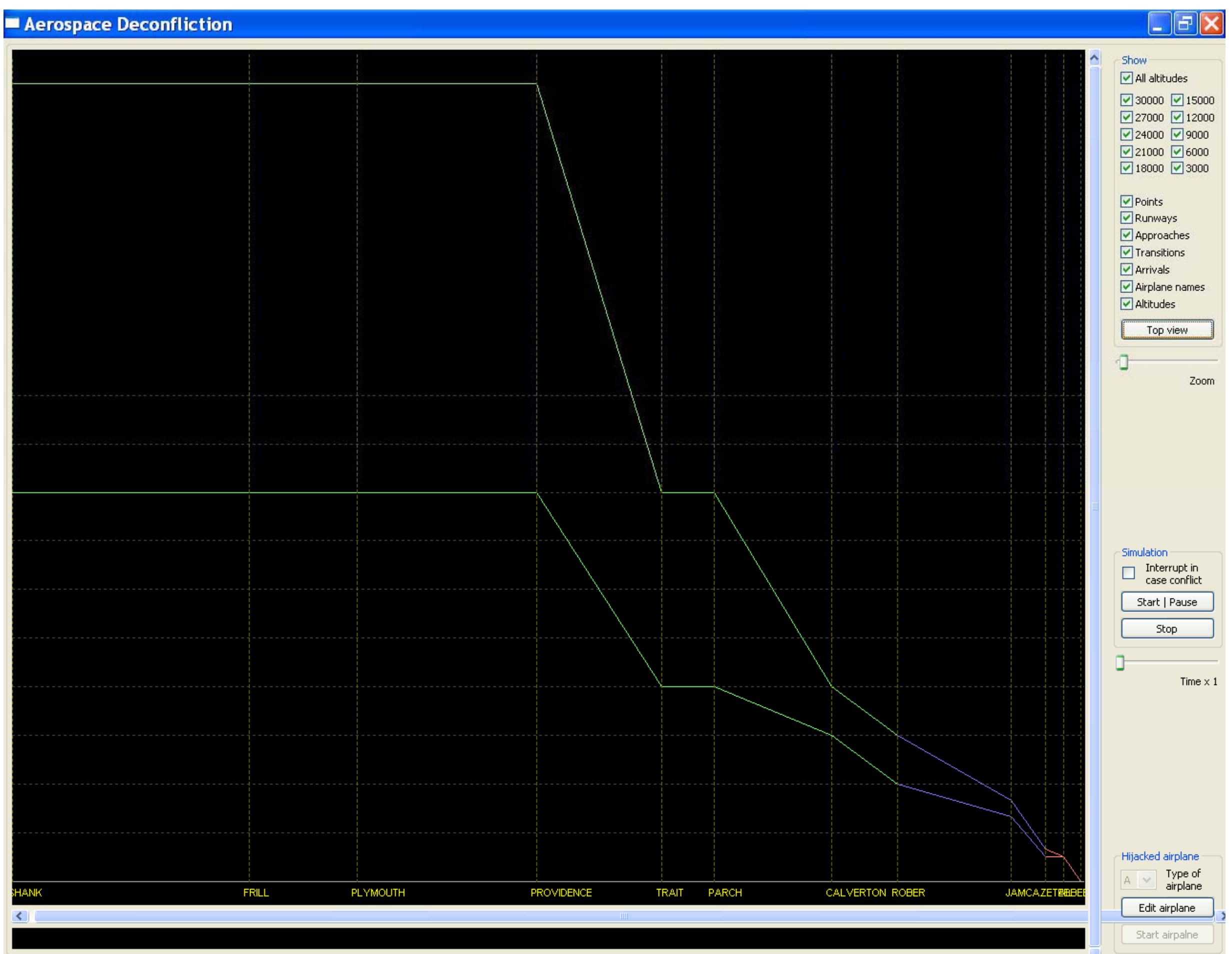


Fig. 6.2. Visualization of movement scheme – related situation in vertical plane

"every third" Γ echelon is depicted in order to make the picture more clear. In particular, 10 echelons are depicted in Fig. 6.2 that correspond to the echelons of the following altitudes: 900 m (3000 feet), 1800 m (6000 feet) etc. till the altitude 9000 m (30000 feet).

Green lines represent boundaries of altitude determining admissible echelons for corresponding legs of the scheme according to specification of the JFK airport airspace topology.

According to the proposed deconfliction model, flexible selection of the echelons is considered as a basic strategy of the deconfliction algorithm. Therefore, this graphical interface may be also considered as a mean for graphical validation of the agents behavior and deconfliction algorithm as a whole.

6.3. Representation of Hijacked Aircraft Trajectory

Trajectory of the hijacked aircraft movement is specified prior the simulation process. This specification is done in two steps. In the first step, its trajectory is specified in horizontal projection (Fig. 6.3). This is done via selection of a sequence of the points of the trajectory. For example, in the example presented in Fig. 6.3 the trajectory is represented by four points.

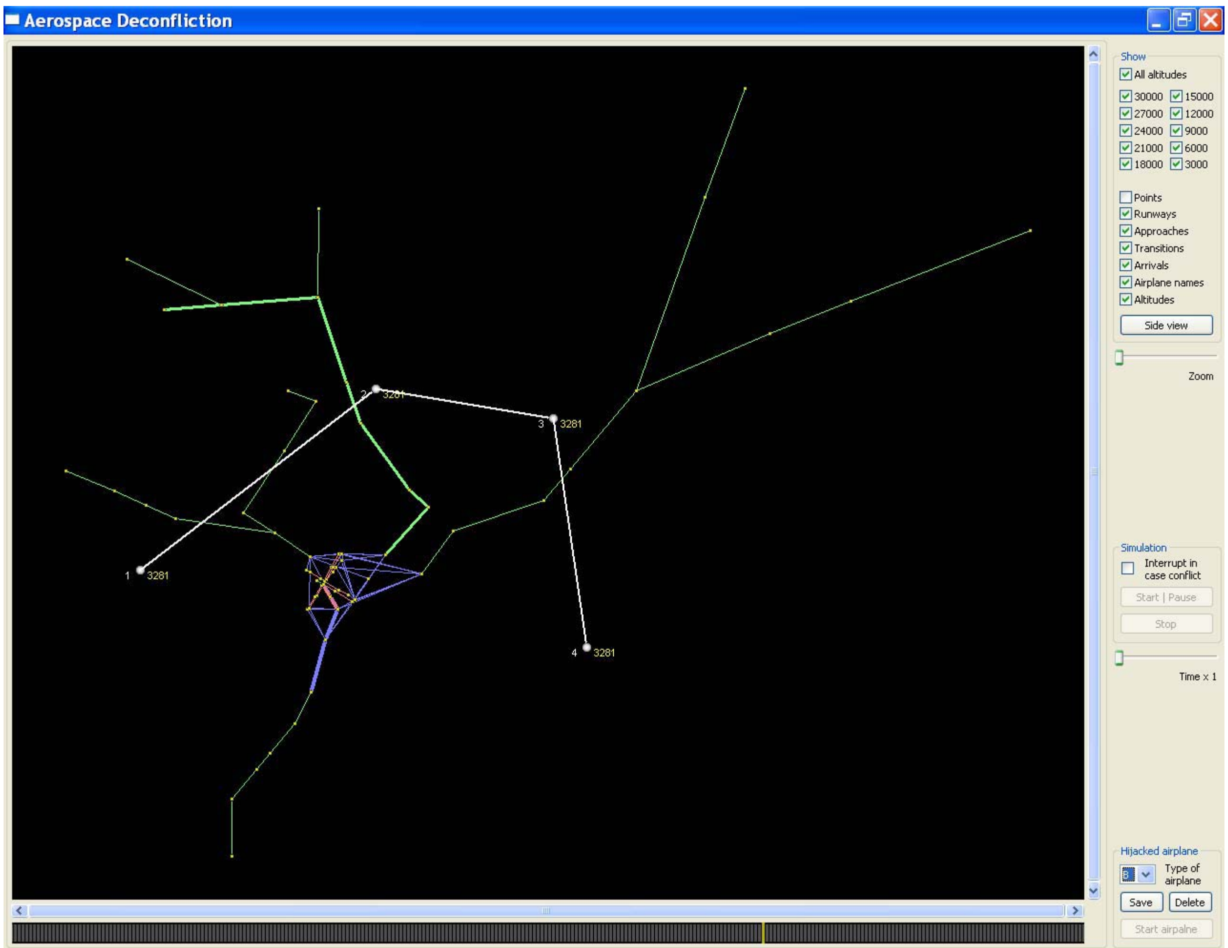


Fig. 6.3. Representation of the hijacked aircraft trajectory in horizontal projection

In the second step, the trajectory is specified in vertical projection. For this purpose the interface depicted in Fig. 6.4 is used. In it, the altitudes of the points selected in the first step are defined.

The time instant corresponding to appearance of the hijacked aircraft is defined manually during the simulation procedure.

To define the hijacked aircraft speed, the aircraft of is assigned a class. Next, depending on the altitude its speed is determined using the speed interval admissible for the particular aircraft class (see Tab. 2.1, section 2.2).

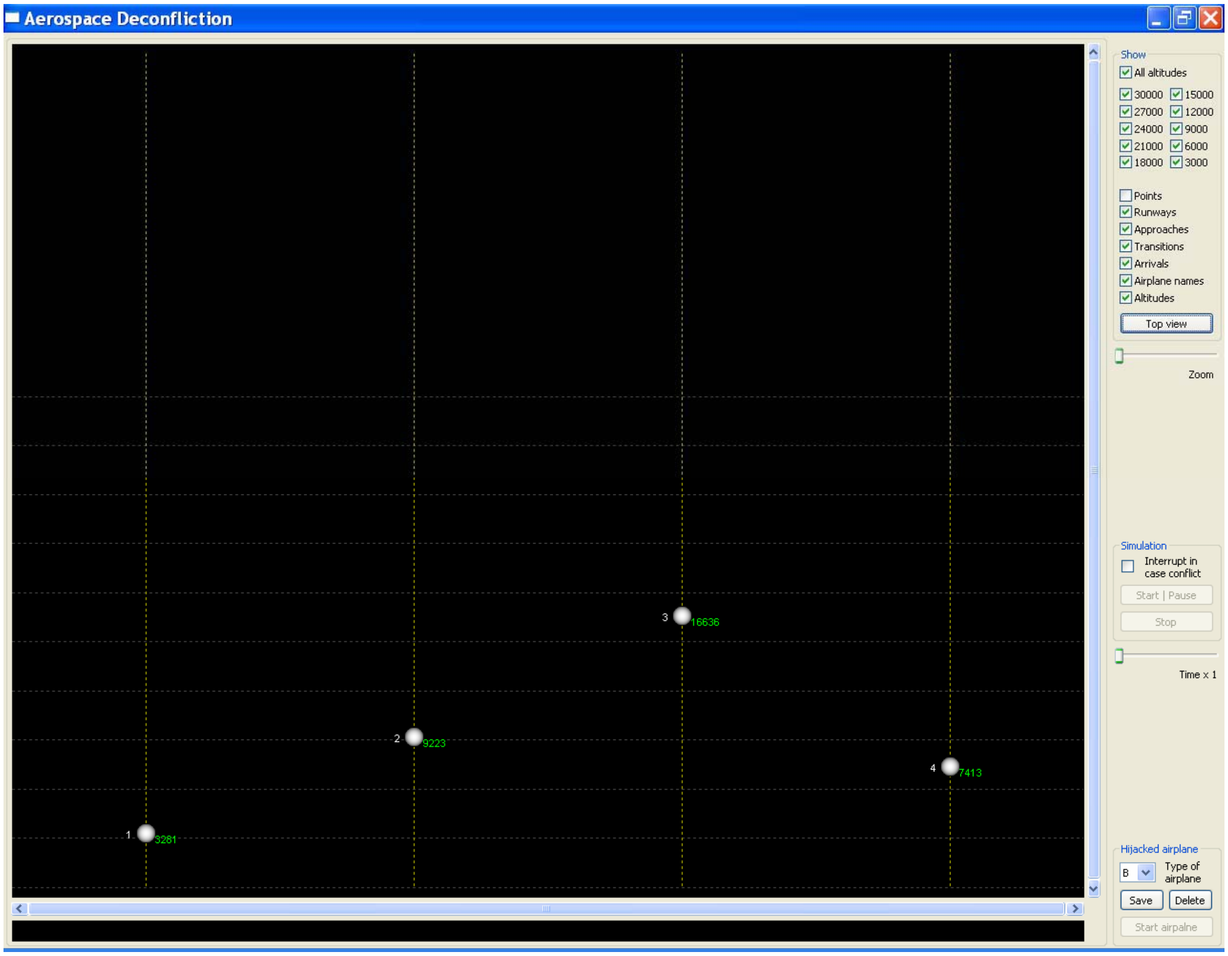


Fig. 6.4. Specification of the trajectory of the hijacked aircraft in vertical projection

6.4. Specification of Initial Air Traffic Related Situation

Specification of particular air traffic situation is based on use of real life timetable of arrival and departure. This is done using graphical user interface. In the developed version, the timetable of the JFK airport of New York City is used.

6.5. Chapter concluding comments

Chapter 6 presents graphical user interface providing visualization of the air traffic configurations and corresponding situations in both, normal situations when only "normal" aircraft operate within airport airspace and abnormal ones, when a hijacked aircraft is operating as well. This interface corresponds to the solution of the Task 7 of the Project Work plan. At the same time, this interface plays the role of important component of the software means supporting verification and validation of the airspace deconfliction algorithm itself.

Report Conclusion

The first basic objective of the current phase of the research is development of about real life model (both conceptual and formal) of airport airspace topology determining admissible movements of aircraft, safety policy (both conceptual and formal) determining existing rules and regulations of safe air traffic within airport airspace and, development and verification, within the above content, airspace deconfliction algorithm. The second basic objective is to develop a conception of distributed airspace deconfliction strategy, using the paradigm of autonomous operation of "normal" aircraft within deconfliction scenarios, which assumes minimal intervention of air traffic operator. This air traffic deconfliction conception should be implemented as a cooperative multi-agent system comprised autonomous intelligent agents assisting the pilot in crisis situations caused by appearance, within airport airspace, hijacked aircraft(s) that does not follow air traffic control rules and air traffic operator commands. The second objective has to be achieved via careful development of the design project of multi-agent airspace deconfliction system containing detailed formal specification of the autonomous pilot assistant agent cooperative behavior within typical scenarios, both, normal and deconfliction.

Both aforementioned objectives are achieved and corresponding results are presented in the Report.

Several important conclusions and recommendations resulting from simulation–based verification of the developed deconfliction model and algorithm are outlined below.

1. The developed deconfliction algorithm is based on creation of the coordinated aircraft' plans focused on analysis of occupied and free echelons within particular sectors of the airport airspace. Effectiveness of deconfliction algorithm depends mainly on two properties of the airport airspace and current air traffic configuration. The first of them is determined as relative intensity (density) of the air traffic within the sectors. In quantitative terms, it is evaluated with the value $d(S)$ that is relation of the count of the aircraft (it is the same as the total count of occupied echelons) of the sector S to the total count of the echelons which are permitted for use at the end point of the sector where aircraft transit into the next sector. It is clear that $d(S) \in [0,1]$. E.g., if $d(S)=1$ that means all the echelons at exit sector point ether occupied or reserved. In such situation the next aircraft is forced to use vectoring maneuver[7].

The second property is determined by a degree of closeness of the airport airspace zone to the approach zone. Formally these characteristics can be evaluated as the value of the following function:

$$t(S)=(n(sh)-n(S))/(n(sh)),$$

where $n(sh)$ –the total count of sectors composing corresponding movement scheme and $n(S)$ is the total count of sectors remaining in the scheme before approach scheme.

These characteristics can be used as effective heuristics for both, "a priory" assessment of the deconfliction task complexity for particular arrival schemes and for use of this information for regulation of the entry of new aircraft into airport airspace and also for assignment of entry points for newly arriving aircraft. An important conclusion on effectiveness of the algorithms (its capability to find satisfactory solution in situations of various air traffic density) is that it is capable to effectively solve deconfliction task within arrival zone is the value of $d(S)$ is not very close to 1. Otherwise, the vectoring maneuver has to be used for deconfliction.

It is worth to note that in real life practice the value of the characteristic $d(S)$ varies within interval 0.2 – 0.3 and extreme situations are practically excluded.

2. It is reasonable to use such deconfliction approach that also well performs in "normal" situations. The proposed deconfliction algorithm is exactly such.

[7] In the current phase of the research, such movement option is not considered so far.

List of Publication

V.Gorodetsky, O.Karsaev, V.Kupin, V.Samoylov. Agent-Based Air Traffic Control in Airport Airspace. Accepted for publications in the Proceedings of International Conference on Intelligent Agent Technology (IAT-07), Silicon Valley, November 2-5, 2007.

V.Gorodetsky, O.Karsaev, V.Skormin, V.Samoylov. Multi-Agent Airspace Deconfliction in Homeland Security Scenario. International Conference "Knowledge Intensive Multi-agent Systems" (KIMAS'2007April–May 2007, 2007.

V.Gorodetsky, O.Karsaev, V.Samoylov, S.Serebryakov. P2P Agent Platform: Implementation and Testing. The AAMAS Sixth International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2007), Honolulu, 2007 pp. 21-32.

References

[1993-Task 1-Addendum 3] Final Report on Addendum 3 to Project #1993P "Multi-agent Technology for Airspace Deconfliction", SPIIRAS, December 2006, 40 pp.

[1993P-Task 1-2003). Final Report on Project #1993P "Autonomous Information Collection, Knowledge Discovery Techniques and Software Tool Prototype for Knowledge-Based Data Fusion", SPIIRAS, September 2003, 113 pp.

[1993P-Task1- Ext 1-2 2005] Final Report on Extensions 1, 2 of Project 1993P. SPIIRAS, May 2005, 105 pp.

[ARINC-424] Draft 2 of Supplement 19 to ARINC Specification 424: Navigation System Data Base. Aeronautical Radio, INC, http://www.arinc.com/aeec

[Hill et all, 2005] Hill, J. C., Johnson, F. R., Archibald, J. K., Frost, R. L., and Stirling, W. C. A Cooperative Multi-Agent Approach to Free Flight.AAMAS-2005, pp. 1083-1090.

[ICAO Doc.4444] ICAO Doc. 4444: Air Traffic Management Fourteenth Edition.2001

[MS Flight SDK] Microsoft Flight 2004 Software Development Kit

[Tozicka et all, 2007] Jan Tozicka, Michael Rovatsos, and Michal Pechoucek. A Framework for Agent-Based Distributed Machine Learning and Data Mining. AAMAS-2007.

[Tomlin et all, 1998] Claire Tomlin, George J. Pappas, and Shankar Sastry. Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems. IEEE Transactions on Automatic Control, Vol.43, No 4, April 1998

[Tumer et all] Kagan Tumer, and Adrian Agogino. Distributed Agent-Based Air Traffic Flow Management. AAMAS-2007 , pp. 330-337

[Airport JFK] http://www.airnav.com/airport/KJFK.